

Biomimetic FPGA-based Spatial Navigation Model with Grid cells and Place cells

Adithya Krishna*, Divyansh Mittal*, Siri Garudanagiri Virupaksha, Abhishek Ramdas Nair, Rishikesh Narayanan, and Chetan Singh Thakur

Abstract

The mammalian spatial navigation system is characterized by an initial divergence of internal representations, with disparate classes of neurons responding to distinct features including location, speed, borders and head direction; an ensuing convergence finally enables navigation and path integration. Here, we report the algorithmic and hardware implementation of biomimetic neural structures encompassing a feed-forward trimodular, multi-layer architecture representing grid-cell, place-cell and decoding modules for navigation. The grid-cell module comprised of neurons that fired in a grid-like pattern, and was built of distinct layers that constituted the dorsoventral span of the medial entorhinal cortex. Each layer was built as an independent continuous attractor network with distinct grid-field spatial scales. The place-cell module comprised of neurons that fired at one or few spatial locations, organized into different clusters based on convergent modular inputs from different grid-cell layers, replicating the gradient in place-field size along the hippocampal dorsoventral axis. The decoding module, a two-layer neural network that constitutes the convergence of the divergent representations in preceding modules, received inputs from the place-cell module and provided specific coordinates of the navigating object. After vital design optimizations involving all modules, we implemented the tri-modular structure on Zynq Ultrascale+ field-programmable gate array silicon chip, and demonstrated its capacity in precisely estimating the navigational trajectory with minimal overall resource consumption involving a mere 2.92% Look Up Table utilization. Our implementation of a biomimetic, digital spatial navigation system is stable, reliable, reconfigurable, real-time with execution time of about 32 s for 100k input samples (in contrast to 40 minutes on Intel Core i7-7700 CPU with 8 cores clocking at 3.60 GHz) and thus can be deployed for autonomous-robotic navigation without requiring additional sensors.

Keywords

Path integration, Autonomous robot navigation, Time-multiplexing, Continuous Attractor Network, Field programmable gate array, Neuromorphic computing.

1 Introduction

Efficient spatial navigation is extremely essential for the survival of most animals and insects as these organisms actively navigate through the environment in search of food, mating partners or favourable surroundings. The spatial navigation system of the brain comprised of multiple brain regions including the hippocampus, the entorhinal and retrosplenial cortices [1]–[8]. These brain regions are equipped with numerous spatially selective cell types such as grid cells [5], [6], place cells [2], [3], head-directions cells [9], [10], boundary-vector or border cells [11], [12] and many more [13], [14]. Specifically, grid cells, which exhibit multiple firing fields in a regular hexagonal or triangular pattern in open spaces, have been implicated in path integration [6], [15]–[20]. Place cells, on the other hand, exhibit single or few firing fields in the explored environment and are considered to encode the current location of organisms with respect to the environment [2], [3], [21]–[26]. Together, interactions among these cells have been postulated to guide the computation of current location of an organism even in the absence of sensory inputs by using self-motion cues [27]–[29]. Of the many models which can produce grid cells firing pattern [17], [18], [30]–[36], experimental studies provide strong lines of evidence in support of a continuous attractor network (CAN) model for explaining the grid cell activity [37]–[39]. Therefore, the CAN model is a widely accepted model to describe the grid cell activities as well as used to further understand the nuances associated with grid cell computation and path integration [39]–[45]. But, being a network model with huge synaptic connectivity, the computation time often becomes unrealistic, especially for large networks. Hence, there is a need to develop real-time computing hardware implementation of network of grid cells and place cells. From the applications standpoint of autonomous robotic navigation, which requires constant update about novel environments, a model for spatial navigation needs to include computations based on both grid cells and place cells. There have been prior attempts for implementing both, the grid cell activity individually or the complete models of spatial navigation on hardware. But, most of the grid cell hardware implementations lack the physiological details present in the brain especially in terms of the emergence of grid cells within a continuous attractor network framework [46]–[57]. Lastly, brain-inspired computation often provides an optimal solution for computational problems such as autonomous robotic spatial navigation. In this study, we report the successful development of a model for spatial navigation based on grid cell activity derived from a CAN model and place

The authors are with the Indian Institute of Science, Bangalore - 560012, India. A. Krishna, S. G. Virupaksha, A. R. Nair and C. S. Thakur are with the Department of Electronic Systems Engineering (Email: {adithyaik, abhisheknair, csthakur}@iisc.ac.in, sirigv.1997@gmail.com; Tel.: +91 9731680833); D. Mittal and R. Narayanan are with the Department of Molecular Biophysics (Email: {divyansh,rishi}@iisc.ac.in; Tel.: +91-80-22933372, Fax: +91-80-23600535).

This work was funded by Pratiksha Trust, Indian Institute of Science and SERB (Science and Engineering Research Board), India.

** These authors contributed equally to this study.*

cell activity derived from the modular inputs from these grid cells. We test this model for decoding efficiency of location tracking in arbitrary arenas and ultimately implement the model employing field programmable gate array (FPGA) based hardware.

Prior hardware implementations have either taken an analog approach [51], [55] or have not implemented the complete models for spatial navigation [51], [53]. Aggarwal [55] built the silicon hippocampal formation with three analog chips, a strip ring chip, a grid cell chip based on the GRIDSmap model and the place chip based on Bayesian integration. Our design utilizes time-multiplexing of neurons and uses a small number of LUTs. The entire design fits on the same FPGA fabric, thereby avoiding off-chip communication. Massoud and Horiuchi [51] implemented an analog VLSI circuit that just generates the hexagonal or triangular grid pattern using continuous attractor dynamics. While analog designs are limited by accuracy, precision, scalability and are not immune to process variations, our digital implementation of the spatial navigation system on FPGA overcomes these drawbacks. Furthermore, our design has the complete model of grid cells, place cells, and a decoding module that is capable of learning new paths and predicting the animal's position in space.

1.1 Our contribution

The work presented here has two significant contributions: (i) We propose a novel, brain-inspired spatial navigation model employing grid cells and place cells, (ii) We develop a hardware architecture to implement the proposed spatial navigation model on FPGA.

To begin with, we present the top-level architecture of the proposed brain-inspired spatial navigation model consisting of a grid cell, place cell, and a decoding module. The grid cell activity is modelled using Continuous Attractor Network (CAN) model. We propose certain enhancements to the current CAN model by introducing a novel step-like ring of inhibitory connectivity to regulate the neuron's outgoing weights. The CAN model with this type of connectivity has never been implemented on hardware before. Compared to the conventional "Mexican-Hat" connectivity, the proposed connectivity has low computational cost and can be easily implemented on hardware. Next, we introduce a place module that integrates modular inputs from grid cells, and the decoding module implemented using a two layer neural network to predict the trajectory explored in an arena using place cell inputs. Our architecture offers the complete spatial navigation model of grid cells, place cells and a decoding module which is not proposed in the prior studies.

We then present the hardware architecture for individual modules of the proposed model. The key features of the proposed architecture are:

- **Low resource utilization:** We leverage on the high speed of modern FPGAs to process large number of neurons without substantially increasing hardware resources, by incorporating time-multiplexing approach. Using this technique, it is not necessary to implement all neurons physically on silicon, instead we implement few neurons on hardware and re-use the same to emulate all neurons in time, thus saving the resource utilization on the hardware. Further the multiplication of the synaptic inputs with synaptic weights are replaced by the shift operation inside the grid cell and place cell modules. This multiplier-less design strategy helps in reducing FPGA resources substantially.
- **Reliability:** Our system is a fully digital design and is largely immune to process variations and device mismatch, which are the key problems faced by large-scale analog and mixed-signal designs.
- **Real-time:** Contrary to software-based implementations, our FPGA-based model can perform computations in real-time, enabling the system to be deployed for robotic navigation application.
- **Flexibility:** Owing to the programmable nature of FPGA the design can be easily reconfigured for different configurations.

We have successfully implemented the proposed spatial navigation model on Zynq Ultrascale+ FPGA. The architectures presented here have contributed to the development of the first fully digital hardware (FPGA) prototype for the spatial navigation using grid cells and place cells. The rest of the paper is organised as follows: We provide the top-level architecture of the proposed model and explain the mathematical formulation of grid cells and place cells in Section 2. The architectures for implementing the proposed model on hardware is provided in Section 3. Sensitivity analysis, evaluation of decoding efficiency and the FPGA resource utilization of the proposed model is provided in Section 4. Section 5 concludes the paper and presents the potential implications of this work and future outlook.

2 Methods and materials

2.1 Virtual trajectories

Simulation and analysis of grid cell activity with a Continuous Attractor Network (CAN) model requires a rat trajectory in a large open field (3×3 m of square arena and 3 m diameter of the circular arena for our analysis). But trajectories recorded from real rat run in an open arena are often very slow and require a lot of time to explore the whole arena. Additionally, running such a long simulation for a huge network with extensive connectivity, such as in the CAN model, becomes very difficult. Hence to reduce the time of simulation and computation, we employed virtual trajectories developed earlier [58] using the following algorithm:

- 1) Rat starts at $(x_0 = 1, y_0 = 1)$ of square or circular arena.
- 2) The initial distance (d_0) covered by the rat is picked from a uniform distribution ($d_0 \in (0, 0.004)$) and the initial angle (A_0) is picked from a uniform distribution ($A_0 \in (0, 2\pi)$).

3) The coordinates of first step of rat movement x_1 and y_1 are calculated by:

$$\begin{aligned}x_1 &= x_0 + d_0 \sin(A_0) \\ y_1 &= y_0 + d_0 \cos(A_0)\end{aligned}$$

4) Afterwards, at each time step, a random variable denoting distance is picked from a uniform distribution ($d_t \in (0, 0.004)$) and another random variable denoting the angle of movement is picked from another uniform distribution of ($A_t \in (\pi/36, \pi/36)$). The new coordinates are updated by:

$$\begin{aligned}x_t &= x_{t-1} + d_t \sin(A_t) \\ y_t &= y_{t-1} + d_t \cos(A_t)\end{aligned}$$

5) If x_t and y_t are more than the bounds of the arena (3×3 m of square arena and 3 m diameter of the circular arena), the d_t and A_t are re-picked until the x_t and y_t are inside the bounds of the arena.

6) If rat is near the boundary of arena, the A_t random variable was picked from a uniform distribution of ($A_t \in (0, 2\pi)$) instead of uniform distribution of ($A_t \in (\pi/36, \pi/36)$). This additional constraint ensured that at the edges of arena there is enough turn in the trajectory to mimic real rat run in an open arena [58].

In a previous study [58], it has been shown that these virtual trajectories closely mimic the real rat trajectories and provides better control over simulation parameters. The need, and preference over completely randomized movement, for these rat-like trajectories arose from the requirement for head direction inputs for grid cell computation [30]. In the absence of such smooth rat-like trajectories, the head direction cells would provide inputs to the grid cells that are drastically changing every timestep of the simulation, thus destabilizing grid cell activity. 20 such virtual trajectories, with different seed values, in open square arena of size 3×3 m and one 3 m diameter circular arena were employed with above given equation and used in this study.

2.2 Top level architecture

There are three distinct modules in this spatial navigation model, namely, the grid cell module, the place cell module and a decoding module as shown in Fig. 1. The input to the model is velocity $V = [V_x, V_y]$, where V_x and V_y are the components of velocity along x and y axes. The grid cell module includes multiple layers of grid cell networks, broadly mapping on to the dorso-ventral axis of the medial entorhinal cortex. Specifically, there is experimental evidence that grid cells in the MEC are arranged in increasing order of grid field size and spacing along the dorso-ventral axis [59]–[62]. To replicate this biological feature in our model, we designed different layers of grid cell network to be endowed with unique grid field size and spacing, with systematic increase in these grid cell properties to reflect the dorso-ventral arrangement in rodents. To accomplish this, we simulated grid cell activity in each of these layers (number of grid cell network layers are five, unless specified) using independent continuous attractor network models with different parametric regimes. It is known that the grid cell macroscopic modules along the dorsoventral axis have increase in grid field size and spacing in progressive ratio of 1.4 to 1.7. We have adjusted the width of inhibitory connection within the grid cell network to maintain this spatial period ratio [63]–[67].

Turning to the place-cell module which represents hippocampal neurons, it is known that similar dorso-ventral modularity exists in size of firing field of hippocampal place cells, with ventral neurons exhibiting larger place field sizes [68]. To mimic this, we imposed modular connectivity from the grid-cell network layers to neurons in the place-cell modules along the dorso-ventral axis. Although neurons within the continuous attractor models of individual grid-cell layers were endowed strong connectivity, neurons within the place-cell modules did not possess recurrent connectivity. Each place cell receives one input from each grid cell layer which ultimately provides input to decoding module. The decoding module was designed as two-layer neural network with one hidden layer and an output layer. The hidden layer was composed of 150 neurons and the output layer consists of 2 neurons to estimate the position $\hat{z} = (\hat{x}, \hat{y})$ in a 2D Cartesian co-ordinate system.

2.3 Grid cell module: CAN model for grid cell activity

Grid cell activity is simulated using a modified version of a 2D continuous attractor model developed by Burak and Fiete [30]. Here, we have explained the CAN model briefly and with appropriate modifications. The rate-based neurons in the network are recipient of inputs from two sources, namely, the feed-forward velocity dependent input and intra-network synaptic inputs. The intra-network synaptic inputs follow centre-shifted ‘‘Mexican Hat’’ or ‘‘centre-surround’’ connectivity in a toroidal network. If we assume the network to be a 2D sheet (open up the toroid), the network size for each grid-cell layer, unless stated otherwise, is $30 \times 30 = 900$ neurons ($n \times n = N$). Each neuron in the network layer is endowed with a preferred direction θ_i or assumed to receive selective inputs from head direction cell with preferred direction θ_i . θ_i can take values of $0, \pi/2, \pi$ and $3\pi/2$ depicting east, north, west and south respectively. These four types of direction specific cells (repeating local block of four cells with each direction) further uniformly tiled the entire network. The classical way to introduce Mexican Hat connectivity is to employ the difference of Gaussians:

$$W_{ij} = W_0(x_i - x_j - l\hat{e}_{\theta_j}) \quad (1)$$

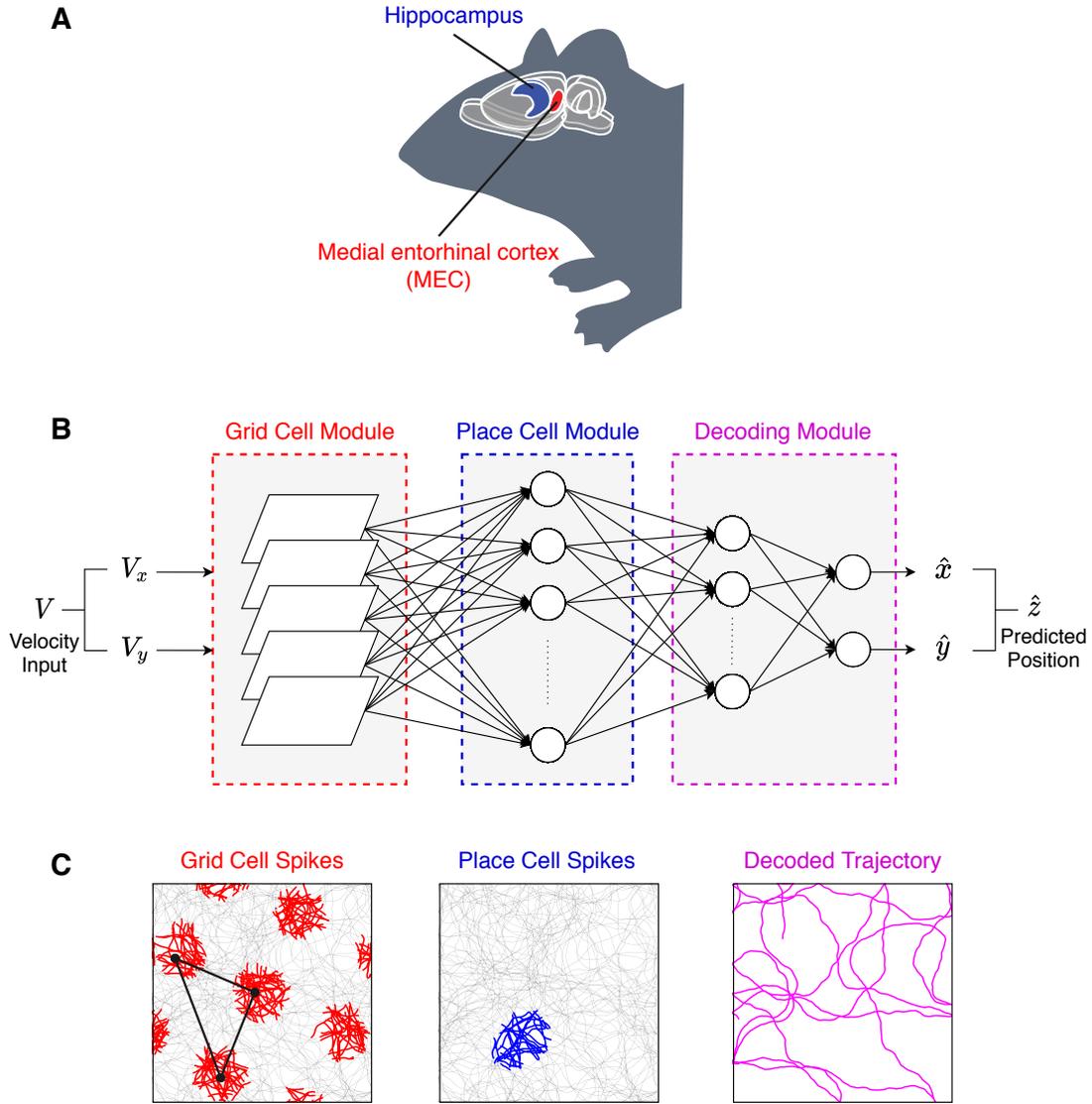


Fig. 1: Top level architecture comprising of the grid-cell, the place-cell and the decoding module. (A) A schematic illustration of rodent brain highlighting the hippocampus (blue) and the medial entorhinal cortex (MEC) (red). (B) Top level architecture for the proposed brain-inspired trimodular model of spatial navigation. The velocity inputs of a virtual rodent or object navigating in an open arena are fed into the first module of the model, a multi-layer, continuous attractor network based, grid-cell module (red, dashed box). The modular outputs from the different layers of grid-cell module are integrated in individual cells of place-cell module (blue, dashed box). A two-layer neural network received inputs from individual cells of the place-cell module and provide the predicted coordinate of navigating virtual rodent or object, is used as decoding module (magenta, dashed box). (C) The example outputs of the three modules are depicted as grid cell spikes (red, left) from a single neuron in the grid-cell module, place cell spikes (blue, centre) from a single neuron in the the place-cell module and the predicted trajectory using decoding module (magenta, right). The original trajectory of navigating rodent or object in an open arena of 3×3 m is shown with grey colour.

where,

$$W_0(x) = ae^{-\gamma|x|^2} - e^{-\beta|x|^2} \quad (2)$$

W_{ij} represents the synaptic weight from neuron j to i . This weight matrix was endowed with center-surround connectivity, with center shifted by $x - l\hat{e}_{\theta_j}$. $a = 1$ ($a = 1$ defines an all-inhibitory network, $a > 1$, suggests excitatory connections in addition to inhibitory connections), $\gamma = 1.1 \times \beta$, $\beta = 3/\lambda^2$, where λ is the periodicity of the lattice in the neural sheet and \hat{e}_{θ_j} defines the unit vector pointing along the θ_j direction. However, as the aim of this study is to implement this type of grid cell computation on a neuromorphic hardware setup, implementing such a difference of Gaussians weight matrix on hardware is computationally very expensive. Hence, we developed a novel abstraction to implement centre-surround

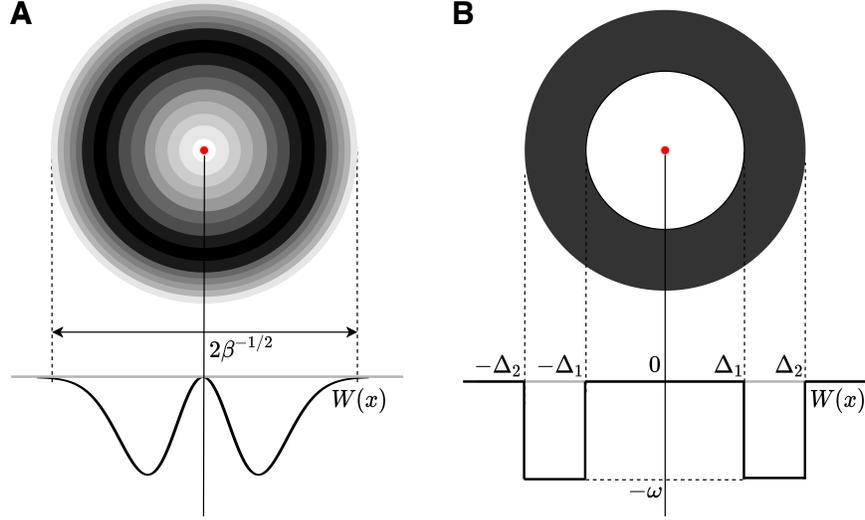


Fig. 2: Network connectivity for emergence of grid cell activity: 2D (top) and 1D (bottom) schematic depiction of the outgoing weights of a neuron in the network. All neurons in the network have same outgoing connectivity pattern except the individual direction preference dependent asymmetry. (A) Mexican-Hat Connectivity with $\beta = 3/\lambda^2$; where λ defines the periodicity of triangular or hexagonal repetitive activity pattern formed on neural lattice. (B) Step-like ring of inhibitory connectivity with $|\Delta_1 - \Delta_2|$ defines the periodicity of triangular or hexagonal repetitive activity pattern formed on neural lattice.

connectivity whereby we replaced the Mexican-Hat weight matrix with a “step-like” ring of inhibitory connectivity:

$$W_0(x) = \begin{cases} -\omega, & \text{if } \Delta_1 \leq |x| \leq \Delta_2. \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

where, ω is the depth of “step-like” ring of inhibitory connectivity and denotes the strength of inhibitory synaptic weight. Δ_1 and Δ_2 denotes the inner and outer radius of ring of inhibition from the center of a cell respectively as shown in Fig. 2. The width of the ring of inhibition is given by $|\Delta_1 - \Delta_2|$. In all simulations and hardware implementation we used, $\omega = 1/32 \approx 0.03$.

Feed-forward velocity-based inputs are computed as:

$$B_i = (1 + \alpha \hat{e}_{\theta_j} V) \quad (4)$$

where, α is a velocity scaling factor, V is the velocity of rodent running and \hat{e}_{θ_j} defines the unit vector pointing along the θ_j direction. If $l = 0$ or $\alpha = 0$, a static repetitive triangular pattern will be formed over the neural lattice. Non zero l and α lead to coupling of velocity with neural dynamics. l defines the shift in the outgoing weights based on Mexican-hat connectivity and is fixed at 2, while the velocity scaling factor (α) is fixed at 60. Following equation describes the dynamics of rate-based neurons for computing both inputs:

$$\tau \frac{ds_i}{dt} + s_i = f \left[\sum_j W_{ij} s_j + B_i \right] \quad (5)$$

where, f is a simple rectification non-linearity representing the neural transfer function such that: $f(x) = x$ for $x > 0$, and is 0 otherwise. s_i denotes the current synaptic activation of neuron i at time point t . τ denotes the time constant of neural response and dt denotes time step for numeric integration. Unless specified the ratio τ/dt was fixed at 16. Activity (S) of all the neurons were recorded over each time step.

The grid cell model employed in this study is a rate-based mean-field continuous attractor network model, which is directly adopted from Burak and Fiete [30]. In this model, the synaptic temporal dynamics is not explicitly modelled, rather at every time step, the synaptic transfer is a scaled version of the neuronal output based on the connectivity weight matrix. Nevertheless, the neuronal output has its own low-pass filtering or integrator temporal dynamics.

Each layer of grid cell CAN model is initialized with randomized input ($S_0 \in (0, 1)$) to all the cells. To ensure the reproducibility of the results, the initial randomized inputs were saved. Initial constant feed-forward drive ($V = 0$ for first 100 input samples) ensured that a stable spontaneous activity pattern is formed over the neural lattice. The output of each neuron in the network will be a grid cell like firing in 2D open arena and will act as input to the place cell module.

2.4 Place cell module: Single place cell dynamics

The dynamics of a place cell is given by:

$$p_k = \sum_{l=1}^L W_k^{(l)} s^{(l)} \quad (6)$$

$W_k^{(l)}$ represents the synaptic weight from a neuron of grid cell layer l to neuron k in the place cell network. $s^{(l)}$ denotes synaptic activation of a neuron in grid cell layer l . The total number grid cell layers utilized for generating place cell response is given by parameter L which is set to five in this work. p_k denotes the activity of place neuron k . 250 neurons are used in the place cell network and each neuron receives one input from each grid layer and it is linearly added to generate the place cell activation. Layers in grid cell module have multiple spatial scales with unique grid field size and spacing. In order to retain multiple spatial scales across place cell network, we divided the place cell neurons into five clusters with 50 neurons present in each cluster. As grid-cell layers and place-cell clusters were employed to mimic dorso-ventral structure, the number of place-cell clusters was chosen to be same as the number of grid layers. To impose modularity of dorso-ventral connectivity, each place-cell cluster was endowed with different grid-to-place synaptic weights such that one of the inputs to the place cell from a single grid layer was assigned higher weight compared to inputs from other layers. Specifically, neurons present in place-cell cluster g receives input with the weight $W_k^{(g)} = 1$ from layer g and input weights from other layers $W_k^{(l \neq g)} = 0.25$. Such modular weight organization reflecting dorso-ventral connectivity aids in endowing our framework to span multiple scales of place field sizes, in a manner that mimics the anatomy and physiology of the entorhinal-hippocampal network.

The grid cell-to-place cell models often consider (e.g., Solstad et al. [28]) that a single place cell receives input from multiple grid cells with heterogeneous properties. In addition, our model accounts for the biological dorso-ventral organization of the grid-cell network, whereby grid cells in the dorsal entorhinal cortex have smaller grid fields compared to those in the ventral entorhinal cortex. However, in many of these grid cell-to-place cell models (e.g., Solstad et al. [28]), the grid phase of all these grid cells projecting to a place cell are considered to be similar, which is not based on physiological or anatomical evidence. To avoid such artificial constraints, we randomized the grid cell selection from each layer, which constitutes an important novelty of our approach demonstrating that convergence of randomized grid-cell inputs (of different scales and grid phase) could yield place-cell firing fields. The details of hardware implementation of place cells is provided in Section 3.2.

2.5 Decoding module: Artificial neural network

Decoding module is implemented using a two layer fully connected neural network with 150 hidden neurons and 2 output neurons. Place cell activations, which are 250 in this case, are fed as the input to the neural network and the network outputs the estimated position of the rodent, $\hat{z} = (\hat{x}, \hat{y})$ in 2D Cartesian co-ordinate system. Activation of hidden layer neurons in a network is given by:

$$a^{[H]} = f^{[H]} \left(W^{[H]} * X + b^{[H]} \right) \quad (7)$$

where, $a^{[H]}$ is a vector containing activation of all hidden layer neurons i.e, $a^{[H]} = [a_1^{[H]}, a_2^{[H]}, \dots, a_{150}^{[H]}]^T$. $a_i^{[H]}$ is the activation of hidden layer neuron i . $f^{[H]}$ is the non-linear activation function employed for hidden layer neurons and we have used Rectified Linear Unit (ReLU) activation function as it can be easily implemented on hardware. $b^{[h]}$ represents the biases of hidden neurons. X is the input vector which contains activation of all place cells i.e, $X = [p_1, p_2, \dots, p_{250}]^T$. $W^{[H]}$ is the weight matrix containing weights from the input layer to the hidden layer and has dimension $[150 \times 250]$.

Activation of output layer neurons is given by:

$$a^{[O]} = f^{[O]} \left(W^{[O]} * a^{[H]} + b^{[O]} \right) \quad (8)$$

where, $a^{[O]}$ is a vector containing activation of output neurons ($a^{[O]} = [a_1^{[O]}, a_2^{[O]}]^T$). $W^{[O]}$ is the weight matrix having weights from the hidden layer to the output layer and has dimension $[2 \times 150]$. $b^{[O]}$ denotes the biases of output neurons. $f^{[O]}$ is the linear activation function employed for the output layer. The final position estimate of the rodent is given by:

$$\hat{z} = [a_1^{[O]}, a_2^{[O]}] = [\hat{x}, \hat{y}] \quad (9)$$

Training of the neural network was done offline using standard back-propagation algorithm [69] and the desired weights and biases obtained are stored in the memory on FPGA. Hardware implementation details of the decoding module is provided in Section 3.3.

2.6 Spatial map analysis of grid cell and place cell activity

Spatial rate maps of grid cell activity were computed by dividing the arena into 100×100 bins. For a particular bin, the activity (s_i) of a specific grid cell (cell i) is added for entire duration of the simulation. This activity is normalized by the probability of finding the rodent in the bin under consideration (occupancy). This process is repeated for all the bins to make the spatial rate maps for an individual grid cell (cell i). Likewise, spatial maps of all the grid cells in each layer were computed. An identical process was employed for computation of the spatial rate map for all place cells using 40×40 bins.

3 Hardware implementation

As mentioned earlier the top level architecture for spatial navigation has three distinct modules namely a grid cell module, a place cell module and a decoding module. A grid cell module is made of multiple layers of grid cell networks responsible for generating grid cell activity with different spatial scales. The output of the grid cell module is fed as the input to the place cell network to generate the place cell activity which is ultimately given to the decoding module to estimate the position of the rodent in a 2D arena. In this section, we provide the hardware architecture and implementation details of grid cell, place cell and decoding modules. The grid cell and place cell modules are implemented using Verilog Hardware Description Language (HDL) and the decoding module is realised using Vivado High-Level Synthesis (HLS).

3.1 Grid cell module

A grid cell module estimates the neuronal outputs of each grid cell of five layers with 900 neurons present in each layer. Thus, the total number of neurons processed inside the grid cell module is $5 \times 900 = 4500$ neurons. Each layer has a different spatial scale with distinct size and spacing of the grid fields. The top level architecture of the grid cell module is shown in Fig. 3. The architecture utilizes four main sub-modules, namely, a connectivity matrix, a temporary buffer, a neuron array and a memory block. A connectivity matrix module stores the indices of other neurons from which a neuron receives its inhibitory input for all 4500 neurons. It is developed using five Read Only Memory (ROM) blocks to store the indices for five layers. A detailed explanation of the connectivity matrix module is provided in Section 3.1.1. A temporary buffer is used to store the current activation states of neurons and it is developed using the block Random Access Memory (RAM) units with each of depth 900. A neuron array is the main computation block consisting of 60 neurons working in parallel to estimate the neuronal activations of the next time step. We exploit the high speed of modern FPGA in our architecture and thus, we do not have to physically implement all 4500 neurons; rather, we only have 60 physical neurons implemented on the hardware. We use the time-multiplexing approach and re-use 60 physical neurons implemented on hardware to estimate activations of 4500 neurons in time. This approach has low resource utilization and consumes less area on digital hardware. A detailed explanation of a single neuron architecture that implements Eq. 5 is provided in Section 3.1.2. The memory module shown in Fig. 3 is used to save the updated neuronal activations of the next time step. It is developed using five block RAM units each of depth 900 to store the activations of five layers.

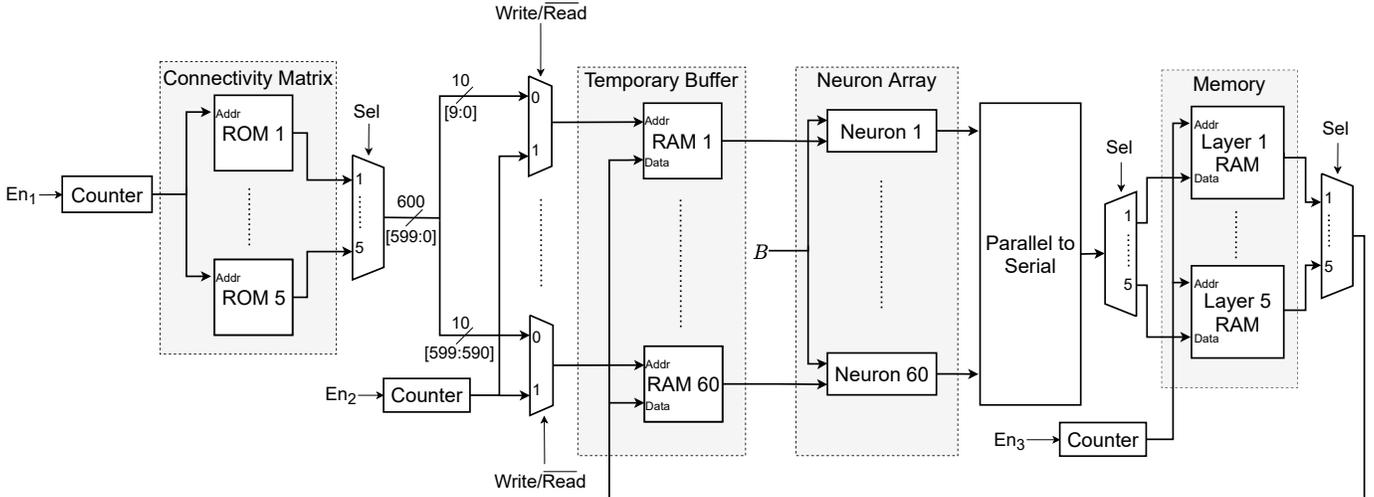


Fig. 3: Grid cell module architecture. The design consists of a connectivity matrix module to store the indices of neurons, a temporary buffer to store current activation states of neurons, a neuron array to perform neural computation given by Eq. 5 and a memory module to save the updated activations of neurons. Sequence of operations performed within the architecture is illustrated in Fig. 4.

A flowchart in Fig. 4 demonstrates the sequence of operations performed within the grid cell module to estimate the neuronal activations. In the first iteration, the contents of each RAM in the memory module are initialized with randomized neurons' state variables. Then the *Sel* (Select) control signal of the multiplexer at the output of the memory module is made 1 to load the contents of layer 1 RAM in the memory module to the temporary buffer. After loading, each block RAM unit in the temporary buffer holds current activation states of 900 neurons present in layer 1. Then the current activation states are updated inside the neuron array with 60 physical neurons parallelly. Each neuron in a neuron array receives a feed forward velocity input (B) and the inhibitory inputs from a certain set of surrounding neurons governed by the step-like ring of inhibitory connectivity. The indices of these specific set of surrounding neurons are provided by the connectivity matrix module. To extract the corresponding activations of these specific surrounding neurons we feed indices obtained

from the connectivity matrix module as the read address to RAMs in the temporary buffer. Since we have 60 neurons working in parallel, we use 60 RAM units in the temporary buffer and perform 60 parallel reads to obtain the current activations of surrounding neurons concurrently. A neuron array updates the current activations of the first 60 neurons, and since we can not store all the 60 values concurrently in RAM inside the memory block, we use a parallel to serial block, which converts parallel inputs to the serial output stream. The parallel to serial block is implemented using a shift register in which the data is simultaneously loaded onto individual registers and then the block of data is sent out serially. The output stream from the parallel to serial block is saved in the first 60 locations of layer 1 RAM in the memory module. In the next stage, we update the activations of the next 60 neurons in the neuron array and save them in the memory module. This process of reading the current activation states from the buffer, updating the activations in a neuron array and saving them back to the memory is repeated for $900/60 = 15$ iterations to the update activation of all 900 neurons in layer 1. After layer 1 activations are updated, we repeat the above procedure from the start to update the activations of other layers. The temporary buffer is loaded only at the start of the computation of a particular layer and it is not necessary to load in every iteration. After completing the computations for all five layers, we use these updated activations stored in the memory module to evaluate the place cell activations inside the place module.

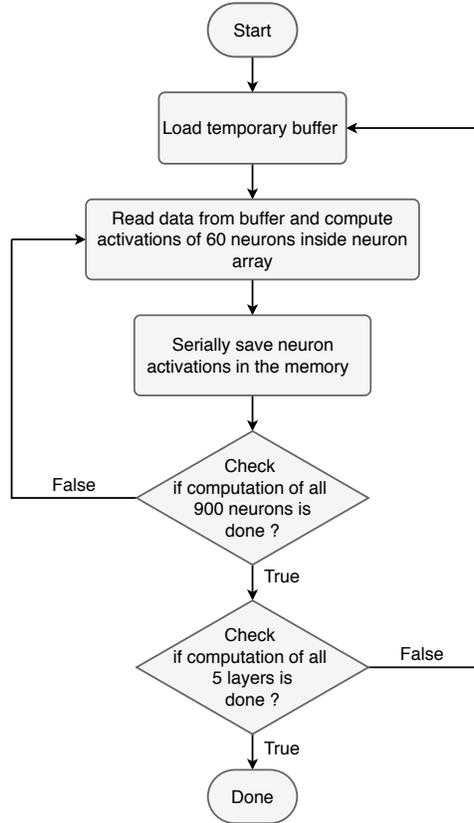


Fig. 4: Flowchart illustrating the sequence of operations carried out by the grid cell module.

3.1.1 Connectivity matrix module

A neuron receives inhibitory inputs from a certain set of surrounding neurons which is governed by a step-like ring of inhibitory connectivity. The indices of these specific set of surrounding neurons are stored in the connectivity matrix module. Since each neuron has its unique input connectivity pattern with other neurons we have to store $m^{(l)} \times 900$ indices for all 900 neurons in a grid cell layer l with $m^{(l)}$ number of inhibitory inputs or connections per neuron. In addition to storing the indices of neurons which provides the input, we also store the index of the receiving neuron for which current activation states are updated. The contents of the single ROM inside the connectivity matrix module is shown in Fig. (5).

The first row of the ROM is used to store the indices of the first 60 neurons and the following $m^{(l)}$ rows are used to store the indices of other neurons, which provides the input to the first 60 neurons. Following which, we store the indices of the next 60 neurons and their corresponding indices of the input neurons in the subsequent rows. In this way, we store the indices of all 900 neurons and their corresponding indices of the input neurons. The total depth of the ROM is $m^{(l)} \times 15 + 15$. 10 bits are used to store the index of a neuron and since we have 60 indices stored in a row the width of the ROM is 600 bits. At every clock cycle, a row is read from the ROM which gives the indices for 60 neurons and these indices are used as the read address of RAMs in the temporary buffer to extract their corresponding current activation states. Since we have five layers with different spatial scales, the connectivity pattern of the neuron and the number of connection or

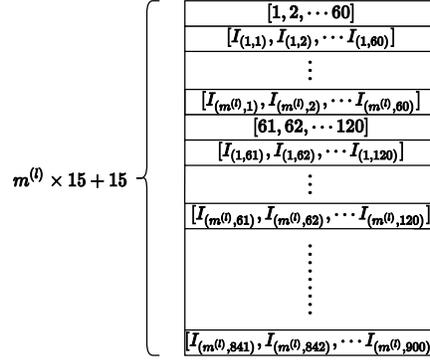


Fig. 5: Contents of a single ROM in the connectivity matrix module.

inputs to a neuron is different for different layers. Thus, we make use of five ROM units in the connectivity matrix module to store the indices of neurons present in each layer separately.

3.1.2 Neuron

The dynamics of a neuron in the grid cell network is given by Eq. 5. By re-arranging the terms, the activation of the neuron i at time step $t + 1$ is updated as,

$$s_{i,t+1} = s_{i,t} + \frac{(s_{inputs} - s_{i,t})dt}{\tau} \quad (10)$$

where, $s_{i,t}$ is the current activation state of the neuron i and s_{inputs} is given by,

$$s_{inputs} = f \left[\sum_j W_{ij} s_{j,t} + B_i \right] \quad (11)$$

For the “step-like” ring of inhibitory connectivity we can rewrite the above equation as,

$$s_{inputs} = f \left[B_i - \sum_j \omega s_{j,t} \right] \quad (12)$$

Thus a neuron i in the grid cell network receive two inputs: the inhibitory synaptic input from other neurons in a grid layer and the feed-forward velocity based input. The architecture of a single neuron i in the neuron array is shown in Fig. (6).

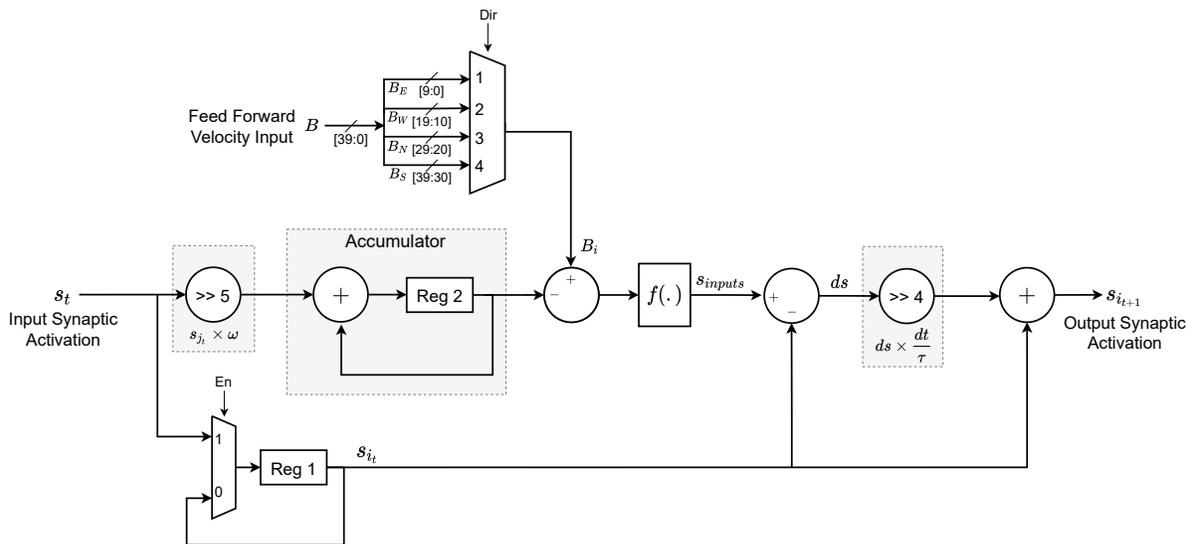


Fig. 6: Single neuron architecture.

Based on the directional preference of the neuron i , a multiplexer is used to select one of the preferred feed-forward inputs B_E , B_W , B_N and B_S depicting east ($\theta_i = 0$), west ($\theta_i = \pi$), north ($\theta_i = \pi/2$) and south ($\theta_i = 3\pi/2$) respectively.

For instance, if a neuron has a preferred direction of North then B_N is selected and given as the input B_i by making Dir (Direction) control signal as 3.

In the first cycle, the current activation of the neuron i whose activation needs to be updated is fetched from the temporary buffer and is stored in the register (Reg 1) by making Enable (En) signal high. In the subsequent cycles, En signal is made low and s_i gives the input activations of other neurons j which is multiplied with the weight ω . ω was set to $1/32$ as the multiplication with $1/32$ can be implemented easily using a simple right shift operator. This saves the area and resource utilized on digital hardware. The right shift operation is denoted by $\gg 5$ in Fig. 6 which indicates that the input is shifted 5 bits to the right, and is mathematically equivalent of multiplying the input with $1/32$. Further, we use an accumulator to accumulate all the inhibitory input from other neurons and then subtracted with the feed-forward velocity input B_i . The output is passed through a simple rectification non-linearity f which is implemented by performing a comparison operation such that, $f(x) = x$ for $x > 0$, and is 0 otherwise. The ratio dt/τ is set to $1/16$ and the multiplication of ds (cf. Fig. 6) with dt/τ is performed using right shift operation which is indicated by $\gg 4$. Finally, the output $s_{i,t+1}$ is stored in the memory block (cf. Fig. 3) of the grid cell module.

3.2 Place cell module

A place cell module uses activations of neurons in the grid cell network to estimate the place cell activations according to Eq. 6. We randomly select the activation of a single neuron ($s^{(l)}$) from each grid cell layer that provides input to a place cell neuron. Thus, each place cell neuron receives one input from each grid layer and they are linearly added to estimate the place cell activation. The architecture of the place module is shown in Fig. (7).

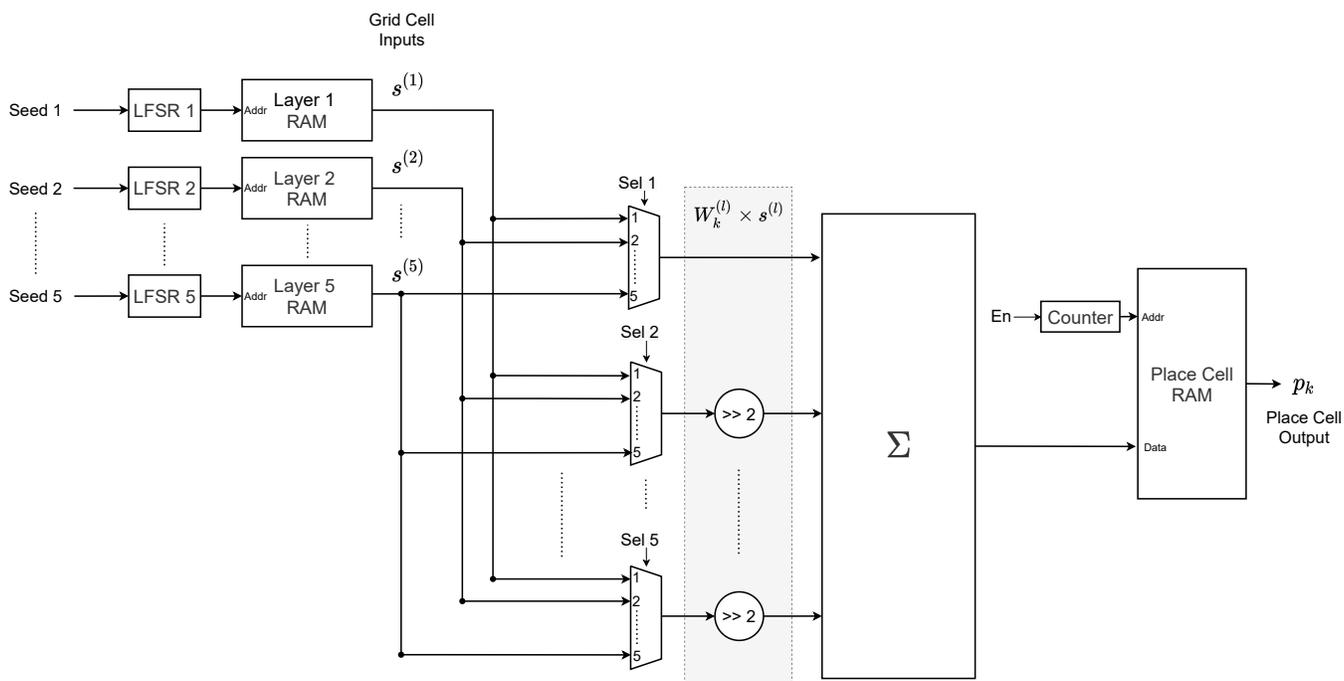


Fig. 7: Place cell module architecture. The grid cell activations from five layers are read and multiplied with their corresponding weights. The output is then linearly added to obtain place cell activations which are stored in the place cell RAM.

To start with, the grid cell activations from five layers stored in their respective RAMs in the memory block of the grid cell module are read in every clock cycle. The read address for the RAMs are provided by a linear feedback shift registers (LFSRs) to randomly pick a neuron from each of the five layers. A detailed explanation of LFSR is provided in Section 3.2.1. The activations retrieved from RAMs are provided as input to the bank of multiplexers. As mentioned earlier, in order to have multiple spatial scales across the place cell network, we divide the neurons in the place network into five clusters with 50 neurons present in each cluster. For estimating the activations of the first 50 place cells belonging to cluster 1, the input weight from layer 1 is set at 1 ($W_{k \in (1,50)}^{(1)} = 1$) and the input weights from other layers are set at 0.25 ($W_{k \in (1,50)}^{(l \neq 1)} = 0.25$) as the multiplication with 0.25 can be implemented easily using right shift operation. For this Sel 1 of the topmost multiplexer is made 1 to feed the activation of layer 1 directly to the adder and Sel (select) lines of other multiplexers are chosen such that they output activations of other layers. The outputs of other multiplexers are multiplied with the weight 0.25 which is implemented using the right shift operation. The output of the topmost multiplexer is fed directly to the adder which is equivalent of multiplying with weight 1 and the output of other multiplexers are right shifted by two positions denoted by $\gg 2$ in Fig. 7 and then fed to the adder which is equivalent of multiplying with weight 0.25. Similarly, for estimating

the activations of place cells present in cluster 2, the input weight from layer 2 is set at 1 ($W_{k \in (51,100)}^{(2)} = 1$) and the input weights from other layers are set at 0.25 ($W_{k \in (51,100)}^{(l \neq 2)} = 0.25$). For this case, the topmost multiplexer is made to select activation of layer 2 by making $Sel = 2$ and the output of other multiplexers are configured to select activations of other layers. In this way, depending on the cluster to which a place neuron belongs, we choose the Sel lines of the multiplexer such that neurons in cluster i receives input from layer i with weight 1 and the input weight from other layers is 0.25. The output of the adder is fed to the place cell RAM to store the place cell activations needed for further computation.

3.2.1 Linear feedback shift register (LFSR)

An LFSR is a shift register that has the feedback mechanism given by combining some of the outputs of the shift register using XOR gates. The feedback points of an LFSR called taps are chosen based on the characteristic polynomial to ensure maximal length sequence. The maximal length sequence of an LFSR gives $2^n - 1$ random numbers before it repeats itself, where n is the length of the shift register. The list of the characteristic polynomial that provides maximal length sequence is given by Alfke [70]. An LFSR is a pseudo random number generator and the output sequence can be determined if the initial seed and the characteristic polynomial are known. As we have five layers, we use five different seeds (cf. Fig. 7) to generate five distinct output sequences. As a single grid layer RAM has the depth 900 to store the activations of 900 neurons in a grid cell layer, we use a 10 bit read address and thus 10-bit LFSR is implemented.

3.3 Decoding module

A decoding module is a simple two layer fully connected neural network with 150 hidden neurons and 2 output neurons. The training of this network is done offline and the desired weights and biases obtained from this training are stored in the random access memory (RAM). As shown in Fig. 8, the place cell activations p_k obtained from the place cell RAM (cf. Fig. 7) forms the input to this module. Based on the layer select multiplexer, the Multiply-Accumulate (MAC) operation for each input generates the output. The first layer input is the place cells and the learned weights which generate the matrix multiplied output in a pipelined manner. The ReLU operation is applied to this output of the first layer using a comparator and now this output is stored in RAM which gives hidden layer activations and forms the input to the second layer. We use the same fully connected layer matrix multiplier hardware for generating the second layer output. Based on the control Sel (select) line, we get the second layer weights and biases from memory and finally generate the decoded output which gives the estimated position of the rodent (\hat{z}). This hardware logic is generated using Vivado High Level Synthesis(HLS) tool.

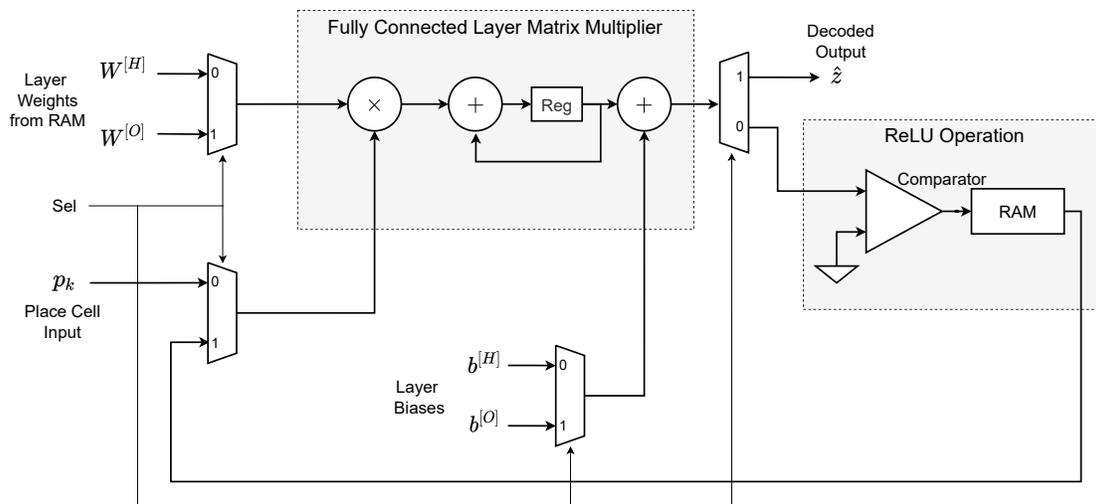


Fig. 8: Decoding module architecture. A decoding module is a two layer neural network. Place cell activations (p_k) forms the input to this neural network. Based on the Sel control signal, corresponding layer weights and biases are selected and Multiply-Accumulate (MAC) operations are performed for each layer. The ReLU operation is carried out at the output of the first layer and the output of the second layer gives the decoded output which denotes the estimated position of the rodent (\hat{z}).

4 Results

4.1 Spatial navigation model

Our spatial navigation model takes inspiration from the brain and comprises of three distinct modules namely the grid-cell, the place-cell and a decoding module. Grid cell activity is simulated using a 2D continuous attractor network model

with some modifications. Traditionally, "Mexican Hat" type of connectivity is employed to govern the outgoing weights of a neuron in the network. But implementing Mexican hat type of connectivity is challenging and computationally very expensive on the hardware, hence we replaced the Mexican hat connectivity with a novel step-like ring of inhibitory connectivity. The juxtaposition of these two type of connectivity is shown in the Fig. (9) by means of 1D and 2D connectivity.

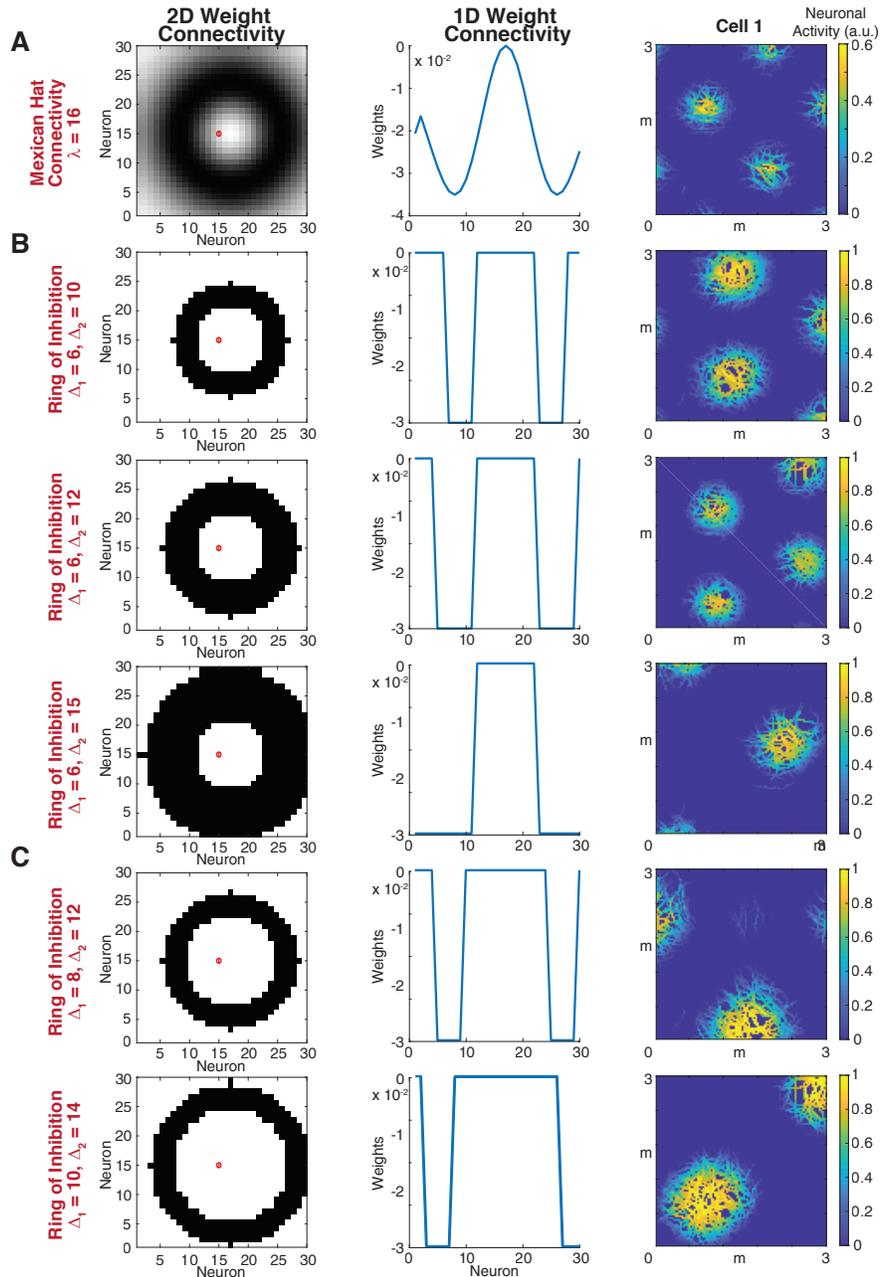


Fig. 9: Substituting "Mexican-Hat" connectivity with "step-like" ring of inhibitory connectivity for grid cell activity using CAN model. Synaptic connectivity and example rate map of grid cell activity using: (A) Mexican-Hat connectivity with $\lambda = 16$; (B) Step-like ring of inhibitory connectivity with fixed $\Delta_1 = 6$ and variable $\Delta_2 = 10$ (1st row), 12 (2nd row), 15 (3rd row) and (C) Step-like ring of inhibitory connectivity with fixed difference between Δ_1 and Δ_2 , $\Delta_1 = 8$ and $\Delta_2 = 12$ (1st row), $\Delta_1 = 10$ and $\Delta_2 = 14$ (2nd row). 1st column: 2-dimensional synaptic weight connectivity matrix for the cell marked with dot in the centre of weight matrix. 2nd column: 1-dimensional slice from the centre of synaptic weight connectivity matrix for cell marked in 1st column. 3rd column: Example rate map obtained using continuous attractor network model.

Each neuron in the grid cell network sends a very strong inhibitory input to the certain set of neurons governed either by Mexican hat or step-like connectivity and the 2D weight representation gives the connectivity matrix for the neuron marked with the red dot in the centre. The direction preference dependent asymmetry in the synaptic weight connectivity, which is essential for continuous attractor model, is evident in Fig. 9. The 1D weight connectivity represents the slice from the centre of the synaptic weight connectivity matrix for the neuron marked with the red dot. Fig. 9 also conveys the effect

of parameters Δ_1 and Δ_2 on size and spacing of the grid fields. From Fig. 9(B) it can be seen that by increasing the width of the ring of inhibition, implemented by keeping Δ_1 constant and increasing Δ_2 , increased the spacing between the grid fields. On the other hand, keeping the width of inhibition constant, implemented by fixing the difference between Δ_1 and Δ_2 , and varying Δ_1 and Δ_2 changed the size of the grid field. This effect of the parameters Δ_1 and Δ_2 on size and spacing of the grid fields aids in bestowing different spatial scales across multiple layers of grid cell networks in the grid cell module.

We then study the effect of network size on the grid cell activity using CAN model with step-like ring of inhibitory connectivity in Fig. 10. The parameters Δ_1 and Δ_2 of the step-like connectivity were set to 7 and 10.5 respectively for generating the rate maps shown in Fig. 10. It can be seen that the minimum network size required for the emergence of grid cell activity is 30×30 neurons as smaller networks did not yield grid cell activity. We consider the network size to be 30×30 for hardware implementation, as it is adequate to establish the grid like behavior and further increasing the network size would increase resource utilization without enhancing performance in terms of grid-cell formation.

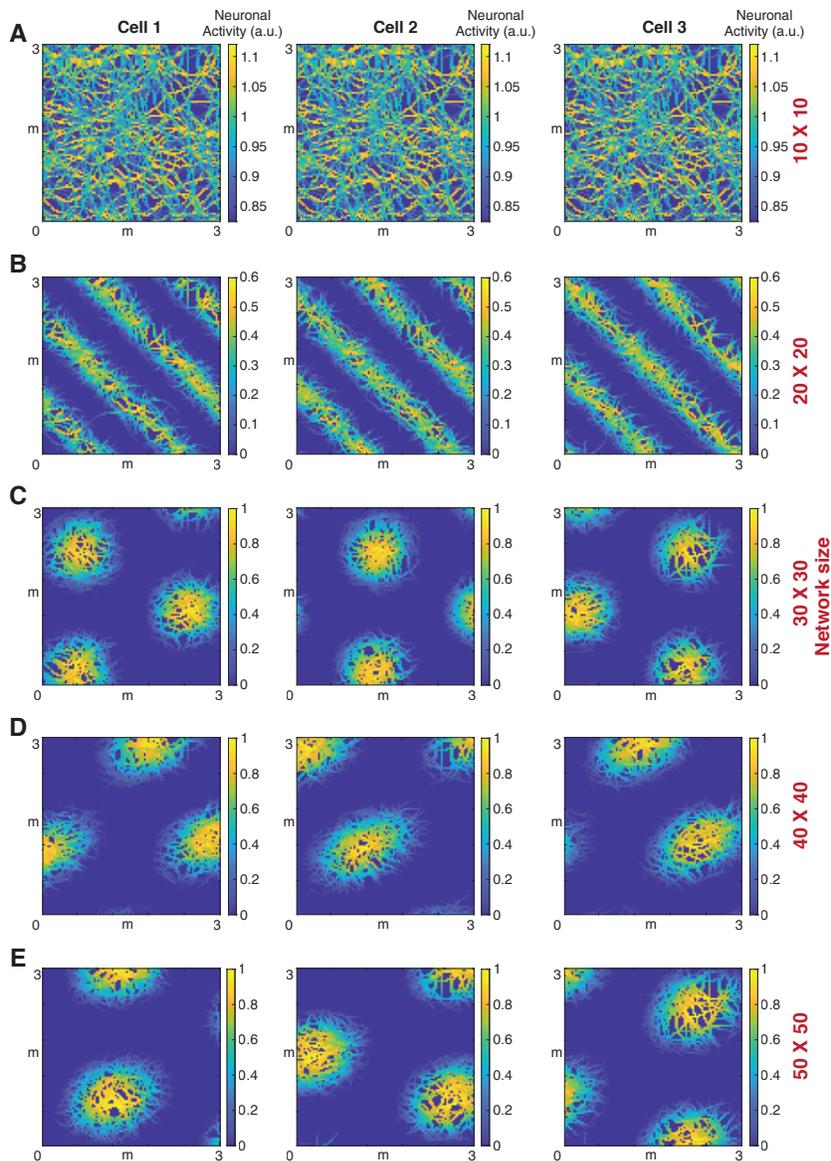


Fig. 10: Minimum network size for emergence of grid cell activity using CAN model with "step-like" ring of inhibition connectivity. Example rate maps of cell 1 – 3 for network size: (A) 10×10 ; (B) 20×20 ; (C) 30×30 ; (D) 40×40 ; (E) 50×50 neurons.

Next, we introduced the place-cell module, a single layer of neurons integrating modular inputs from different layers of grid cell networks, and the decoding module which is a neural network trained to predict the trajectory explored in the arena, to complete our model of spatial navigation. As part of sensitivity analysis, we analyzed the impact of number of layers of grid cell network and the number of place cells on decoding efficiency. The assessment of decoding efficiency was made by plotting the mean squared error (MSE) between the original virtual trajectory and the predicted trajectory as a

function of number of layers and number of place cells as shown in Fig. 11. The MSE between the original trajectory and the predicted trajectory is defined as follows:

$$MSE = \frac{1}{K} \sum_{i=1}^K (z_i - \hat{z}_i)^2$$

where, z_i is the position corresponding to the original trajectory and \hat{z}_i is the position corresponding to the predicted trajectory returned by the decoding module and K is the number of samples used for evaluating MSE.

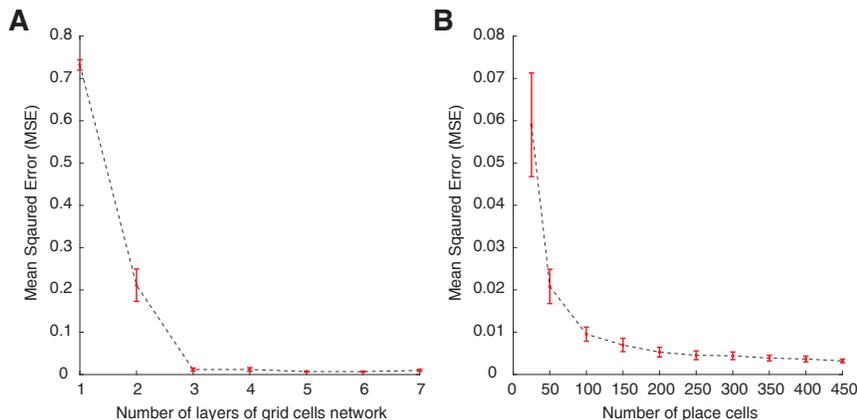


Fig. 11: Decoding efficiency for the grid cell-place cell model for spatial navigation. Mean squared error (MSE) between the original virtual trajectory and predicted trajectory is plotted as function of: (A) Layers of grid cell networks in grid cell module and (B) Number of place cells in the place cell module. For (A), the number of place cells in place module are fixed at 250 neurons while for (B), the number of grid cell network in grid cell module are fixed at five layers. The hidden layer of decoding module had 150 neurons. $K = 30k$ samples are used for calculating MSE. MSE values plotted in both (A) and (B) are the mean of errors over 50 iterations and the error bar denotes the standard deviation.

For Fig. 11(A), we estimated the MSE by varying the number of layers in the grid cell module and fixing the number of place cells in place module at 250 neurons. It can be seen that by adding additional layers in the grid cell module the MSE reduces and to achieve a optimum decoding efficiency we use five layers in our model. For Fig. 11(B), we estimated the MSE by varying number of place cells and fixing the number of layers in the grid cell module at five. The MSE reduces with increase in the number of place cells and we used 250 place cells in our model and for hardware implementation to achieve a optimum decoding efficiency. The MSE values are the average error over 50 iterations and the error was evaluated for $K = 30k$ samples. The error bar in the plot denotes standard deviation.

4.2 Output of each module of the spatial navigation model

The output of each module in the proposed spatial navigation model is shown in Fig. 12. The rate maps of the grid cell activity of five layers with different spatial scales is given by Fig. 12(A). Here we show the rate maps of three adjacent grid cells. Multiple spatial scales are obtained by changing the parameters Δ_1 and Δ_2 for each layer. For hardware implementation, the parameters (Δ_1, Δ_2) were fixed at $(6.5, 9.5)$, $(7, 10.5)$, $(9.5, 12.5)$, $(11, 14)$, $(13.5, 18.5)$ for five layers respectively. The output of grid cell module is given as the modular input to the place cell module and the rate maps of the place cells are shown in Fig. 12(B). Neurons in the place cell network are divided into five clusters with 50 neurons present in each cluster and we show a rate map of a single neuron present in each cluster. The output of the place cells are given as the input to the decoding module to predict the position (\hat{z}). Decoding is implemented using a neural network with 150 hidden neurons and 2 output neurons. Neural network was trained on 10 virtual trajectories in square arena of size 3×3 m and tested on 10 other virtual trajectories in square arena and 1 trajectory in circular arena with diameter 3m. Out of 11 other trajectories tested we show the decoding result for five trajectories in square arena and 1 trajectory in circular arena in Fig. 12(C).

4.3 FPGA implementation

The proposed spatial navigation model was implemented on Zynq UltraScale+ (xczu9eg-ffvb1156-2-e) field programmable gate array. The resource utilization of the implemented design is summarized in Table 1.

All memory blocks used for storing the indices of neurons, activations of grid cells, place cells, weights and biases of the neural network for decoding are mapped into embedded 18kb block random access memory (BRAM) resources available on FPGA, using the Block Memory Generator (BMG) IP [71] provided by Xilinx. BRAMs shown in Table 1 indicates the

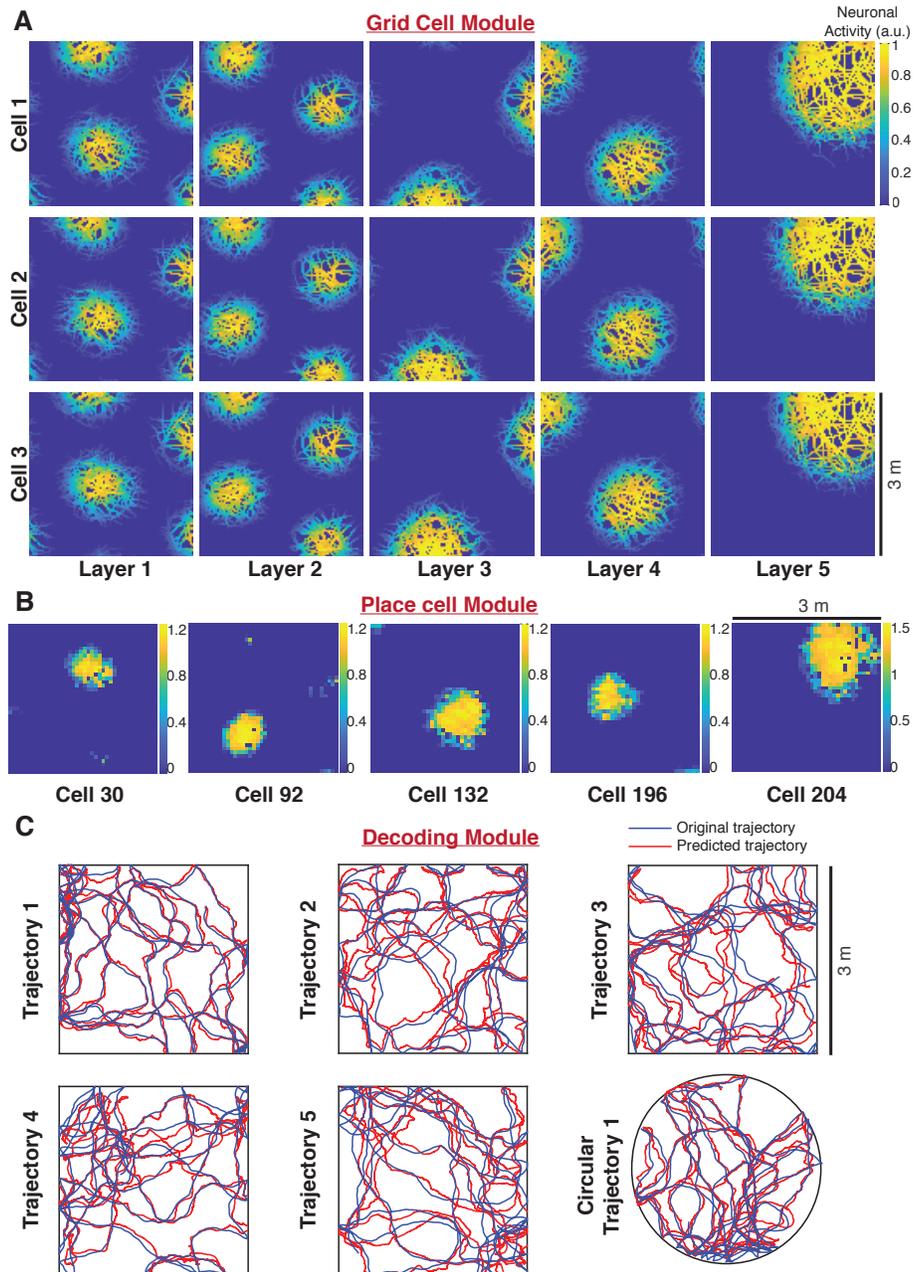


Fig. 12: Output of each module of the grid cell-place cell model for spatial navigation: (A) Grid cell module output: Example grid cell activity rate maps of cell 1 – 3 (rows 1st to 3rd respectively) for each of the five layers. The ring of inhibition parameters in layer 1: $\Delta_1 = 6.5$ and $\Delta_2 = 9.5$ (1st column), layer 2: $\Delta_1 = 7$ and $\Delta_2 = 10.5$ (2nd column), layer 3: $\Delta_1 = 9.5$ and $\Delta_2 = 12.5$ (3rd column), layer 4: $\Delta_1 = 11$ and $\Delta_2 = 14$ (4th column) and layer 5: $\Delta_1 = 13.5$ and $\Delta_2 = 18.5$ (5th column); (B) Place cell module output: Example outputs of five place cells with different field size. (C) Decoding Module Output: neuronal network with 250 neurons in input layer and 150 neurons in hidden layer is trained with 10 virtual trajectories in square arena of size 3×3 m and tested for 11 virtual trajectories (10 additional trajectories in square arena and 1 trajectory in circular arena with diameter of 3 m). Six examples (five trajectories in square arena and one trajectory in circular arena) are shown for both original (blue) and predicted (red) trajectories for $K = 30k$ samples.

number of 18kb BRAM tiles needed on FPGA. Moreover, it can be seen that utilization of LUTs (LookUp Tables), registers and DSPs (Digital Signal Processing) units are very low. Time-multiplexing and several design optimizations such as, replacement of multiplication with shift operation incorporated into the architecture are the primary factors responsible for reduced resource consumption on FPGA.

For implementing the model on FPGA we have converted all the internal variables from a floating point to a fixed point representation. The activations of grid cells and place cells use 10 bit fixed point representation. Weights and biases of the neural network used in the decoding module is 16 bit wide. Finally the output of the decoding module uses 16 bit fixed

TABLE 1: Device utilization summary

Resource	Used	Available	Utilization(%)
LUTs	7,994	274,080	2.92
Registers	5,007	548,160	0.91
BRAMs	477	912	52.30
DSPs	3	2,520	0.12

point representation.

The total computational time (T) of the model implemented on FPGA is given by:

$$T = (L_g + L_p + L_d)T_{clk} \quad (13)$$

where, L_g , L_p and L_d are the number of clock cycles needed for the computation of grid cell, place cell and decoding module respectively. T_{clk} is the clock frequency of the design. In our implementation $L_g = 25,630$ cycles, $L_p = 255$ cycles and $L_d = 37,800$ cycles. Using the clock frequency of 200 MHz the total computational time (T) is 0.318 ms. The input is sampled by the design at the rate $1/T = 3.14$ kHz. Thus the implementation of the proposed model on FPGA makes spatial navigation possible in real-time.

Grid cell and place cell modules were implemented using Verilog HDL and the decoding module was realized using Vivado HLS. The hardware logic generated from both were integrated and then implemented on FPGA. Fig. 13 shows the output of the final decoded result obtained from FPGA. The output is shown for square arena of size 3×3 m. We then compare the MSE of the hardware based result which uses fixed point representation with that of the software result which uses 32 bit floating point representation and conclude that there is no significant difference in the MSE values between software and FPGA based hardware result.

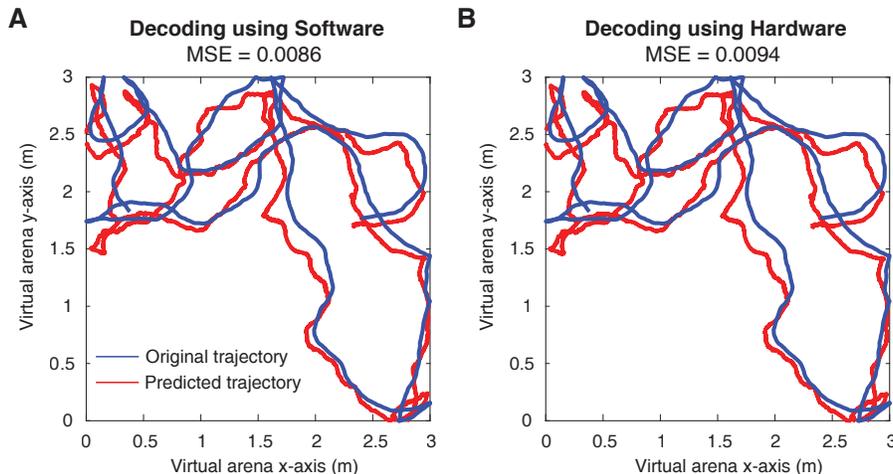


Fig. 13: Comparison of decoding efficiency of software-based and hardware-based results. Example in square arena of size 3×3 m for both original (blue) and predicted (red) trajectories using: (A) Software model and (B) FPGA-based hardware. Note that there is no substantial difference between MSE values for software model result and FPGA-based hardware result. $K = 30k$ samples are used for calculating the MSE.

5 Discussion

We proposed a spatial navigation model inspired by the brain, comprising of grid cells, place cells and a decoding module. The grid cell activity was modelled using multiple continuous attractor network (CAN) models, representing the dorso-ventral span of the medial entorhinal cortex, with some adaptations to make the system feasible and easily realizable on hardware. We replaced the conventional "Mexican hat" connectivity with a novel "step-like" inhibitory connectivity to govern CAN models to aid in having simpler hardware design with reduced computational effort. We demonstrated that the size and position of the inhibitory annulus governed grid cell spacing within the CAN framework, and effectively employed this in defining different grid-cell layers mimicking the dorso-ventral brain organization. In an effort to minimize computational requirements, we employed the minimum network size of CAN models for the effective emergence of grid cell activity. Importantly, we employed clusters of place cells, with defined connectivity from different grid-cell layers, to impose dorso-ventral place cell organization within our framework.

We allowed the propagation of activity from velocity to the different grid-cell layers, then to the place-cell clusters in a modular fashion and finally fed the place-cell output to a decoding module to estimate the position of a virtually navigating animal. We evaluated the decoding efficiency of the proposed model by varying number of layers of grid cell networks and the number of place cells in the place cell network to estimate the optimum grid cell layers and place cells required to achieve a desired decoding efficiency. We implemented our array of the three modules — endowed with optimum number of grid cells with differential annular inhibitory connectivity across different grid-cell layers, optimal number of neurons in the place-cell and the decoding modules, — in hardware on Zynq Ultrascale+ FPGA and demonstrated that the hardware implementation faithfully reproduced the algorithmic results. We established the resource utilization summary of our hardware implementation, and demonstrate that the implementations of our architecture accurately estimated the instantaneous position in 2D arena.

In this study, we aimed to develop a path integrator model using grid cells and place cells firing, with place cells in the hippocampus acting as the spatial output. There are multiple advantages of using place cell output as the decoding layer input. First, our model closely follows the biologically observed connectivity pattern and field size of the grid cells and the place cells along the dorsoventral axis of hippocampal anatomy. Second, the individual place cells provide a unique identity to each location of the explored area and significantly improved the decoding efficiency. Lastly, there is a very strong drift in the grid cell pattern while using the continuous attractor network as a mechanism for grid cell emergence. This drift cause progressive decrease in decoding efficiency as time of exploration is increased. Using the place cell output for decoding bypasses this problem and provides very accurate decoding of spatial exploration even for longer duration, which is further supported by Fig. 11-13. As observed in Fig. 11(A), the decoding with one layer of grid cells is very inaccurate, suggested by the larger error between predicted and original trajectory and hence further support the idea of a grid cell-place cell model for odometric path integration.

From a broader perspective, a common design principle that spans several biological systems is the initial divergence in the neural code, characterized by the presence of disparate classes of neurons responding to distinct attributes of sensory stimuli. This initial divergence is then followed by a representational convergence that provides precise outputs mediating well-defined interactions with the external world. As example would be the visual system, which processes visual information in a piecemeal fashion, with different circuit components involved in processing features including orientation, motion, color, depth, spatial frequency, with eventual convergence (“binding”) that results in a coherent percept of the external visual stimulus.

In the context of our study, the mammalian spatial navigation system is also characterized by such an initial divergence of internal representations, with distinct classes of neurons responding to features including location, speed, borders, and head direction; an ensuing convergence finally enables navigation and path integration. As opposed to the mechanisms underlying the initial divergence of neural representations, the neural mechanisms underlying the eventual convergence of different features are not well-understood. More specifically, the question of how the eventual convergence results in a unified motor output (in the case of the spatial navigation system, say) or in a unified percept (with reference to sensory systems) remains an open and active research problem (the “binding” problem). The decoding module in our model is an “black-box” approximation to this final convergence, and employs the backpropagation algorithm in a multi-layer neural network to provide a solution. We employed the backpropagation algorithm for approximating this black box because of the proven utility of the algorithm in providing such approximation, and given recent arguments on the biological plausibility of the backpropagation algorithm, derived from our current understanding of active dendritic physiology and synaptic plasticity [72].

As one of the major aims of this study is to develop a hardware implementation of spatial navigation system of the brain using grid and place cells models, many simplifications and abstractions are implemented, ranging from single neuron computation to connectivity between grid cells networks and place cells. Physiologically, at single neuron level, grid cells are found in medial entorhinal cortex region of the brain [5], [6]. These cells possess many characteristic properties such as frequency selectivity for the inputs, intrinsic peri-threshold theta frequency oscillations, theta bursting behaviour, mixed-mode oscillations and many more [73]–[80]. These electrophysiological properties are associated with specific set of ion channels present in the neurons [73], [80]. Biophysically realistic conductance-based models of such neurons and small networks can be developed with current computational advances. But continuous attractor network models for grid cell firing require a very large amount computation across hundreds of neurons. As a result, instead of conductance-based models, dynamical abstractions are often used for neural computation. In this study, we used one such abstraction called mean-field modelling approach for CAN model, where, each neuron act as an simple integrator of synaptic inputs [30]. Next, the ring of inhibition or Lincoln-Hat connectivity between the neurons in a grid cell network layer is a simplification of Mexican-Hat connectivity [39], [81], [82]. Yet the grid cells activity with these different connectivity are not quantitatively or qualitatively different [81], [82]. Again, this is done to ensure a quick implementation of connectivity weight matrix onto the hardware. Ring of inhibition resulted in a binary connectivity matrix which drastically reduces the computation requirement on hardware.

One of the major concerns regarding the use of CAN is that, till now, there is no physiological study confirming the direct presence of such attractor in the medial entorhinal cortex of the brain. Nevertheless, studies have suggested presence of low dimensional attractor in the MEC using neuronal population analysis [83]. Furthermore, intracellular recording from grid cells have shown membrane potential signatures similar to what an attractor model will yield [37]–[39]. The CAN model is one of the two prominent competing models that are available in the grid cell literature, with the other

model involving an oscillatory interference paradigm to explain grid-patterned firing [30], [32], [33]. There is an ongoing debate on the abilities of either class of models in explaining grid-cell network physiology. In addition, there are a class of hybrid models, that incorporate components of both continuous attractor network and oscillatory interference models, that have proposed towards better explaining the in vivo intracellular membrane potential behavior during grid cell firing [37], [38], [43]. Experimental evidence point to the absence of intracellular intrinsic peri-threshold membrane potential oscillations and the presence of ramp-like depolarisation as the animal enters the grid field [37], [38]. These lines of evidence provide support for the emergence of grid cell firing through a hybrid CAN-like mechanism, with oscillations providing a supporting role [37], [38]. In conclusion, the CAN model approach for obtaining grid cell activity is the most widely accepted approach and the presence of such low dimension continuous attractor has also been physiologically proven in other model system such as *Drosophila*'s head direction system [84], hence the presence of such low dimensional yet complex attractor state in MEC region of brain is not surprising. Lastly, the grid cell network used here is all inhibitory network, but the grid cell activity of such network is not qualitatively different from an excitatory-inhibitory network where excitatory neurons are connected predominantly through inhibitory synapses [39], [44], [81].

There are contrasting theories and experiments both supporting and questioning the grid cells to place cells transformation and vice-versa [28], [29], [85]–[97]. Although there are increasing lines of evidence for such grid cell to place cell transformation [89], [98]–[102], the major issue with such theories is the absence of direct connections between MEC layer II grid cells to place cell in CA1. Pure grid cells are found in layer II/III of MEC while conjunctive grid cells (head direction modulated grid cells) are found in layer III and V of MEC [16]. The CA1 place cells in hippocampus only receive inputs from Layer III of MEC not from Layer II. While the inputs from layer II of MEC are projected to CA1 neurons via DG and CA3 region of the hippocampus [103]. The direct connection between the grid and place cells in this study are made keeping in mind to reduce the increasing complexity of the model and further difficulty in implementing it to the hardware. Lastly, the place cells receive peri-somatic inhibitory inputs which are missing from the model. The constant inhibitory inputs will further increase the spatial selectivity of the place cells by reducing the excitatory activity of the neurons in the secondary place fields [104], [105]. Future studies should therefore endeavor to implement the entire hippocampal formation, with these connectivity differences accounted for.

In terms of the complete model of spatial navigation, the model used here have only grid cells and place cells while in the brain other cells such as border cells [11], [12], object cells, extra hippocampal place cells [106], [107], goal cells [108], [109] and conjunctive cells [16], [110] are also present [13]. Other than these, the head direction cells [9], [10] and speed cells [111], [112] are not explicitly modelled rather the movement based inputs from these cells to grid cells are modelled [30]. Future studies can accommodate such spatially selective cells to further increase the spatial navigation capability of computational models.

There are many previous attempts for developing spatial navigation system on hardware using distinct network architectures. Simultaneous localization and mapping (SLAM) models use competitive attractor network (a network similar to the continuous attractor network model of grid cell firing) for path integration and simple spatial tasks [113]. Along with 2D spatial exploration, SLAM models are further developed for 3D exploration and path integration [114]. Furthermore, using attractor dynamics of head direction system of the rodent brain and vision-driven reset, process of path integration has been attempted on a chip [53]. In this study, we proposed a novel network architecture for implementation of spatial navigation system for path integration, involving multiple layers of grid-cell networks, place-cells layer and convergent decoding layer. Due to linear transformation of multiple grid cell output to place cell, the network behaves very efficiently, especially compared to a single layer of continuous attractor network for grid cells Fig. 11(A). Future studies can be more focused on comparing these distinct network architectures of task efficiency, computational load and power consumption.

Very few works in the literature have attempted to implement the brain-inspired spatial navigation models on hardware. Prior hardware implementations have either adopted an analog approach or have not incorporated complete models for spacial navigation. Massoud and Horiuchi [51] developed an analog VLSI circuit for modeling a population of grid-cells using continuous attractor dynamics and fabricated a chip with 16×16 array of cells using $0.5 \mu\text{m}$ CMOS process. In other work [52], they implemented a head direction cell system relying on spiking neurons with attractor dynamics and the neuron chip was fabricated using $0.5 \mu\text{m}$ CMOS process. Kreiser et al. [53] realized a head-direction spiking neural network on a neuromorphic device. The neural network received the visual input from the event based camera and used this information to correct the spiking neurons estimate of the robot's orientation. They used the neuromorphic processor Reconfigurable OnLine Learning System (ROLLS) comprising of hybrid analog/digital circuits to emulate the biological processes of spiking neurons and synapses with ultra low-power consumption. Aggarwal [55] designed the silicon hippocampal formation with three analog chips, a strip ring chip, a grid cell chip based on the GRIDSmap model and the place chip based on Bayesian integration and fabricated them using $0.5 \mu\text{m}$ ON semiconductor technology. These prior analog designs are constrained by accuracy, precision, scalability, whereas our digital implementation of the spatial navigation system on FPGA overcomes these limitations. Moreover, the development and testing of large analog implementations is difficult compared to digital systems due to the lack of standard design and test flows. Furthermore, previous hardware implementations realize only a specific sub-module and do not propose an entire system for spatial navigation, on the other hand our architecture has a complete model of grid cells, place cells and a decoding module capable of exploring new path and predicting the location in 2D arena. Although our FPGA based systems gets around the non-idealities of analog VLSI circuits and largely immune to process variations and device mismatch, it will consume more power compared to other analog implementations. The total on-chip power consumption of our system on FPGA is

0.86 W with dynamic power and static power consumption being 0.234 W and 0.625 W respectively.

Previous works in the literature have shown Artificial Neural Network (ANN) based spatial navigation systems [115]–[118], however very few report FPGA implementation [119], [120]. Azhar and Dimond [119] developed an FPGA based adaptive neural controller for robot navigation using ANN. Torres et al. [120] proposed a neural network architecture and its implementation on FPGA for navigation of Unmanned Aerial Vehicles (UAVs). As compared to conventional neural network models, our model is based on neurophysiological properties of the hippocampal cells, thus the parameters of the feature extraction stage (formation of grid and place cells) are fixed, and no training is required in the feature extraction stage. This has an immense advantage over conventional machine learning based models, which require computationally demanding training procedures and suffer in the case of data unavailability. Furthermore, the additional complications due to overfitting, lack of generalization, sensitivity to spurious inputs (as in adversarial attacks) of traditional models defined by tens of thousands or more parameters are well documented [121], [122]. Training these complex deep neural networks is also very costly, in terms of time, energy and other resources.

One important limitation of our hardware implementation is the memory consumption. The connectivity matrix that stores the connections of 4500 grid cells takes up 46.83% of the available BRAMs. This could be a potential problem for scalability. One solution could be taking advantage of the low LUT consumption and including a module to calculate the indices of the connected neurons on the fly without storing them in memory. Another option is to store the connectivity matrix in the DRAM and cache the connections of the set of active neurons in BRAM. In order to save memory and enhance the scalability of the model, a small collection of repeated connectome pattern mimicking the Lincoln-Hat connectivity can be used, and one such example is the canonical axonal fan-in circuit.

The proposed spatial navigation model was developed foreseeing to serve two primary objectives. First, the engineering objective, our brain-inspired spatial navigation model provides the framework for the development of robotic platforms that could perform path integration tasks without the need for an external sensor such as GPS. We suggest that our model could be deployed on robotic platforms for real-time computation, since the computational time of our implementation is extremely low. Second, the scientific objective, to serve as a digital neuromorphic simulator. Recently, there has been a rising demand in developing a large scale neuromorphic simulators [123], [124] and our FPGA based system can serve that purpose for biological understanding of the model's components. Intel Core i7-7700 CPU with 8 cores clocking at 3.60 GHz take about 40 minutes to simulate just five layers of grid cells with 100k input samples, whereas our dedicated hardware takes less than 32 seconds to compute the output of five grid cells layers, place cells layer and the decoding module together. This is really helpful when we need to tune and test these models for better biological understanding. In addition to studying the grid cells and place cells, our design with certain modifications, can also be used to study and understand other parts of the hippocampus that are modelled using the attractor network models like the dentate gyrus and Cornu Ammonis (CA3).

Acknowledgements

This work is funded by Pratiksha Trust, Indian Institute of Science, Ministry of Human Resource Development (MHRD), Science and Engineering Research Board (SERB), India (ECR/2017/002517) and IMPRINT Grant (IMP/2018/000550) from the Department of Science and Technology, India. They would also like to thank their colleague Sathyaprakash Narayanan for his help and support during this work.

Declarations of interest

The authors declare no conflict of interest.

References

- [1] E. C. Tolman, "Cognitive maps in rats and men," *Psychol Rev*, vol. 55, no. 4, pp. 189–208, 1948. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pubmed/18870876>
- [2] J. O'Keefe and J. Dostrovsky, "The hippocampus as a spatial map. Preliminary evidence from unit activity in the freely-moving rat," *Brain Res*, vol. 34, no. 1, pp. 171–5, 1971. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pubmed/5124915>
- [3] J. O'Keefe and D. H. Conway, "Hippocampal place units in the freely moving rat: why they fire where they fire," *Exp Brain Res*, vol. 31, no. 4, pp. 573–90, 1978. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pubmed/658182>
- [4] R. U. Muller, J. L. Kubie, and R. Saypoff, "The hippocampus as a cognitive graph (abridged version)," *Hippocampus*, vol. 1, no. 3, pp. 243–6, 1991. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pubmed/1669298>
- [5] M. Fyhn, S. Molden, M. P. Witter, E. I. Moser, and M. B. Moser, "Spatial representation in the entorhinal cortex," *Science*, vol. 305, no. 5688, pp. 1258–64, 2004. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pubmed/15333832>
- [6] T. Hafting, M. Fyhn, S. Molden, M. B. Moser, and E. I. Moser, "Microstructure of a spatial map in the entorhinal cortex," *Nature*, vol. 436, no. 7052, pp. 801–6, 2005. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pubmed/15965463>
- [7] R. A. Epstein, W. E. Parker, and A. M. Feiler, "Where Am I Now? Distinct Roles for Parahippocampal and Retrosplenial Cortices in Place Recognition," *Journal of Neuroscience*, vol. 27, no. 23, pp. 6141–6149, 2007. [Online]. Available: <https://www.jneurosci.org/content/27/23/6141>
- [8] R. Czajkowski, B. Jayaprakash, B. Wiltgen, T. Rogerson, M. C. Guzman-Karlsson, A. L. Barth, J. T. Trachtenberg, and A. J. Silva, "Encoding and storage of spatial information in the retrosplenial cortex," *Proceedings of the National Academy of Sciences*, vol. 111, no. 23, pp. 8661–8666, 2014. [Online]. Available: <https://www.pnas.org/content/111/23/8661>

- [42] H. Pastoll, L. Solanka, M. C. van Rossum, and M. F. Nolan, "Feedback inhibition enables theta-nested gamma oscillations and grid firing fields," *Neuron*, vol. 77, no. 1, pp. 141–54, 2013. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pubmed/23312522>
- [43] D. Bush and N. Burgess, "A hybrid oscillatory interference/continuous attractor network model of grid cell firing," *J Neurosci*, vol. 34, no. 14, pp. 5065–79, 2014. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pubmed/24695724>
- [44] O. Shipston-Sharman, L. Solanka, and M. F. Nolan, "Continuous attractor network models of grid cell firing based on excitatory-inhibitory interactions," *J Physiol*, vol. 594, no. 22, pp. 6547–6557, 2016. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pubmed/27870120>
- [45] J. Widloski and I. R. Fiete, "A model of grid cell development through spatial exploration and spike time-dependent plasticity," *Neuron*, vol. 83, no. 2, pp. 481–495, 2014. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pubmed/25033187>
- [46] P. J. Zeno, S. Patel, and T. M. Sobh, "Review of Neurobiologically Based Mobile Robot Navigation System Research Performed Since 2000," *Journal of Robotics*, vol. 2016, p. 8637251, 2016. [Online]. Available: <https://doi.org/10.1155/2016/8637251>
- [47] A. Jauffret, N. Cuperlier, and P. Gaussier, "From grid cells and visual place cells to multimodal place cell: a new robotic architecture," *Frontiers in Neurorobotics*, vol. 9, p. 1, 2015. [Online]. Available: <https://www.frontiersin.org/article/10.3389/fnbot.2015.00001>
- [48] G. Wyeth and M. Milford, "Spatial cognition for robots," *IEEE Robotics & Automation Magazine*, vol. 16, no. 3, pp. 24–32, 2009.
- [49] M. J. Milford, J. Wiles, and G. F. Wyeth, "Solving navigational uncertainty using grid cells on robots," *PLoS computational biology*, vol. 6, no. 11, pp. e1000995–e1000995, 2010. [Online]. Available: <https://pubmed.ncbi.nlm.nih.gov/21085643https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2978698/>
- [50] M. Milford and G. Wyeth, "Persistent Navigation and Mapping using a Biologically Inspired SLAM System," *The International Journal of Robotics Research*, vol. 29, no. 9, pp. 1131–1153, 2009. [Online]. Available: <https://doi.org/10.1177/0278364909340592>
- [51] T. M. Massoud and T. K. Horiuchi, "A neuromorphic VLSI grid cell system," in *2012 IEEE International Symposium on Circuits and Systems (ISCAS)*, Conference Proceedings, pp. 2421–2424.
- [52] —, "A neuromorphic head direction cell system," in *2009 IEEE International Symposium on Circuits and Systems*, Conference Proceedings, pp. 565–568.
- [53] R. Kreiser, M. Cartiglia, J. N. P. Martel, J. Conradt, and Y. Sandamirskaya, "A Neuromorphic Approach to Path Integration: A Head-Direction Spiking Neural Network with Vision-driven Reset," in *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, Conference Proceedings, pp. 1–5.
- [54] K. R. Hedrick and K. Zhang, "Megamap: flexible representation of a large space embedded with nonspatial information by a hippocampal attractor network," *Journal of Neurophysiology*, vol. 116, no. 2, pp. 868–891, 2016. [Online]. Available: <https://doi.org/10.1152/jn.00856.2015>
- [55] A. Aggarwal, "Neuromorphic VLSI realization of the hippocampal formation," *Neural Networks*, vol. 77, pp. 29–40, 2016. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0893608016000125>
- [56] H. Tang, W. Huang, A. Narayanamoorthy, and R. Yan, "Cognitive memory and mapping in a brain-like system for robotic navigation," *Neural Networks*, vol. 87, pp. 27–37, 2017. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0893608016301861>
- [57] M. Mokhtar, D. M. Halliday, and A. M. Tyrrell, *Hippocampus-Inspired Spiking Neural Network on FPGA*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, vol. 5216, pp. 362–371. [Online]. Available: http://link.springer.com/10.1007/978-3-540-85857-7_32
- [58] D. Mittal and R. Narayanan, "Resonating neurons stabilize heterogeneous grid-cell networks," *bioRxiv*, 2020. [Online]. Available: <https://www.biorxiv.org/content/early/2020/12/11/2020.12.10.419200>
- [59] V. H. Brun, T. Solstad, K. B. Kjelstrup, M. Fyhn, M. P. Witter, E. I. Moser, and M. B. Moser, "Progressive increase in grid scale from dorsal to ventral medial entorhinal cortex," *Hippocampus*, vol. 18, no. 12, pp. 1200–12, 2008. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pubmed/19021257>
- [60] M. Fyhn, T. Hafting, M. P. Witter, E. I. Moser, and M. B. Moser, "Grid cells in mice," *Hippocampus*, vol. 18, no. 12, pp. 1230–8, 2008. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pubmed/18683845>
- [61] D. L. Garden, P. D. Dodson, C. O'Donnell, M. D. White, and M. F. Nolan, "Tuning of synaptic integration in the medial entorhinal cortex to the organization of grid cell firing fields," *Neuron*, vol. 60, no. 5, pp. 875–89, 2008. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pubmed/19081381>
- [62] L. M. Giocomo, E. A. Zilli, E. Fransen, and M. E. Hasselmo, "Temporal frequency of subthreshold oscillations scales with entorhinal grid cell field spacing," *Science*, vol. 315, no. 5819, pp. 1719–22, 2007. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pubmed/17379810>
- [63] E. I. Moser, M.-B. Moser, and Y. Roudi, "Network mechanisms of grid cells," *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, vol. 369, no. 1635, pp. 20120511–20120511, Dec 2013, 24366126[pmid]. [Online]. Available: <https://pubmed.ncbi.nlm.nih.gov/24366126>
- [64] L. Kang and V. Balasubramanian, "A geometric attractor mechanism for self-organization of entorhinal grid modules," *eLife*, vol. 8, p. e46687, aug 2019. [Online]. Available: <https://doi.org/10.7554/eLife.46687>
- [65] C. Barry, R. Hayman, N. Burgess, and K. J. Jeffery, "Experience-dependent rescaling of entorhinal grids," *Nature Neuroscience*, vol. 10, no. 6, pp. 682–684, Jun 2007. [Online]. Available: <https://doi.org/10.1038/nn1905>
- [66] H. Stensola, T. Stensola, T. Solstad, K. Frøland, M.-B. Moser, and E. I. Moser, "The entorhinal grid map is discretized," *Nature*, vol. 492, no. 7427, pp. 72–78, Dec 2012. [Online]. Available: <https://doi.org/10.1038/nature11649>
- [67] J. Krupic, M. Bauza, S. Burton, C. Barry, and J. O'Keefe, "Grid cell symmetry is shaped by environmental geometry," *Nature*, vol. 518, no. 7538, pp. 232–235, Feb 2015. [Online]. Available: <https://doi.org/10.1038/nature14153>
- [68] M. W. Jung, S. I. Wiener, and B. L. McNaughton, "Comparison of spatial firing characteristics of units in dorsal and ventral hippocampus of the rat," *J Neurosci*, vol. 14, no. 12, pp. 7347–56, 1994. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pubmed/7996180>
- [69] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, Oct 1986. [Online]. Available: <https://doi.org/10.1038/323533a0>
- [70] P. Alfke, "Efficient shift registers, LFSR counters, and long pseudo-random sequence generators," *Tech Notes*, pp. 1–6, 1996.
- [71] "Block Memory Generator v8.4," *Xilinx, LogiCORE IP Product Guide*, pp. 1–129, 2017.
- [72] T. P. Lillicrap, A. Santoro, L. Marris, C. J. Akerman, and G. Hinton, "Backpropagation and the brain," *Nature Reviews Neuroscience*, vol. 21, no. 6, pp. 335–346, Jun 2020. [Online]. Available: <https://doi.org/10.1038/s41583-020-0277-3>
- [73] H. Pastoll, H. L. Ramsden, and M. F. Nolan, "Intrinsic electrophysiological properties of entorhinal cortex stellate cells and their contribution to grid cell firing fields," *Front Neural Circuits*, vol. 6, p. 17, 2012. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pubmed/22536175>
- [74] A. Alonso and E. Garcia-Austt, "Neuronal sources of theta rhythm in the entorhinal cortex of the rat. II. Phase relations between unit discharges and theta field potentials," *Exp Brain Res*, vol. 67, no. 3, pp. 502–9, 1987. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pubmed/3653312>
- [75] A. Alonso and R. Klink, "Differential electroresponsiveness of stellate and pyramidal-like cells of medial entorhinal cortex layer II," *J Neurophysiol*, vol. 70, no. 1, pp. 128–43, 1993. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pubmed/8395571>
- [76] R. Klink and A. Alonso, "Ionic mechanisms for the subthreshold oscillations and differential electroresponsiveness of medial entorhinal cortex layer II neurons," *J Neurophysiol*, vol. 70, no. 1, pp. 144–57, 1993. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pubmed/7689647>
- [77] A. Alonso and R. R. Llinas, "Subthreshold Na⁺-dependent theta-like rhythmicity in stellate cells of entorhinal cortex layer II," *Nature*, vol. 342, no. 6246, pp. 175–7, 1989. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pubmed/2812013>
- [78] A. Boehlen, U. Heinemann, and I. Erchova, "The range of intrinsic frequencies represented by medial entorhinal cortex stellate cells extends with age," *J Neurosci*, vol. 30, no. 13, pp. 4585–9, 2010. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pubmed/20357109>

- [79] A. Boehlen, C. Henneberger, U. Heinemann, and I. Erchova, "Contribution of near-threshold currents to intrinsic oscillatory activity in rat medial entorhinal cortex layer II stellate cells," *J Neurophysiol*, vol. 109, no. 2, pp. 445–63, 2013. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pubmed/23076110>
- [80] D. Mittal and R. Narayanan, "Degeneracy in the robust expression of spectral selectivity, subthreshold oscillations, and intrinsic excitability of entorhinal stellate cells," *Journal of Neurophysiology*, vol. 120, no. 2, pp. 576–600, 2018, pMID: 29718802. [Online]. Available: <https://doi.org/10.1152/jn.00136.2018>
- [81] E. I. Moser, Y. Roudi, M. P. Witter, C. Kentros, T. Bonhoeffer, and M. B. Moser, "Grid cells and cortical representation," *Nat Rev Neurosci*, vol. 15, no. 7, pp. 466–81, 2014. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pubmed/24917300>
- [82] D. C. Rowland, Y. Roudi, M. B. Moser, and E. I. Moser, "Ten Years of Grid Cells," *Annu Rev Neurosci*, vol. 39, pp. 19–40, 2016. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pubmed/27023731>
- [83] K. Yoon, M. A. Buice, C. Barry, R. Hayman, N. Burgess, and I. R. Fiete, "Specific evidence of low-dimensional continuous attractor dynamics in grid cells," *Nat Neurosci*, vol. 16, no. 8, pp. 1077–84, 2013. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pubmed/23852111>
- [84] S. S. Kim, H. Rouault, S. Druckmann, and V. Jayaraman, "Ring attractor dynamics in the Drosophila central brain," *Science*, vol. 356, no. 6340, pp. 849–853, 2017. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pubmed/28473639>
- [85] J. B. Hales, M. I. Schlesiger, J. K. Leutgeb, L. R. Squire, S. Leutgeb, and R. E. Clark, "Medial entorhinal cortex lesions only partially disrupt hippocampal place cells and hippocampus-dependent place memory," *Cell Rep*, vol. 9, no. 3, pp. 893–901, 2014. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pubmed/25437546>
- [86] Y. Dordek, D. Soudry, R. Meir, and D. Derdikman, "Extracting grid cell characteristics from place cell inputs using non-negative principal component analysis," *Elife*, vol. 5, p. e10094, 2016. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pubmed/26952211>
- [87] A. H. Azizi, N. Schieferstein, and S. Cheng, "The transformation from grid cells to place cells is robust to noise in the grid pattern," *Hippocampus*, vol. 24, no. 8, pp. 912–9, 2014. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pubmed/24866281>
- [88] B. R. Kanter, C. M. Lykken, D. Avesar, A. Weible, J. Dickinson, B. Dunn, N. Z. Borgesius, Y. Roudi, and C. G. Kentros, "A Novel Mechanism for the Grid-to-Place Cell Transformation Revealed by Transgenic Depolarization of Medial Entorhinal Cortex Layer II," *Neuron*, vol. 93, no. 6, pp. 1480–1492 e6, 2017. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pubmed/28334610>
- [89] C. Miao, Q. Cao, H. T. Ito, H. Yamahachi, M. P. Witter, M. B. Moser, and E. I. Moser, "Hippocampal Remapping after Partial Inactivation of the Medial Entorhinal Cortex," *Neuron*, vol. 88, no. 3, pp. 590–603, 2015. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pubmed/26539894>
- [90] D. Lyttle, B. Gereke, K. K. Lin, and J. M. Fellous, "Spatial scale and place field stability in a grid-to-place cell model of the dorsoventral axis of the hippocampus," *Hippocampus*, vol. 23, no. 8, pp. 729–44, 2013. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pubmed/23576417>
- [91] T. Neher, A. H. Azizi, and S. Cheng, "From grid cells to place cells with realistic field sizes," *PLoS One*, vol. 12, no. 7, p. e0181618, 2017. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pubmed/28750005>
- [92] D. Bush, C. Barry, and N. Burgess, "What do grid cells contribute to place cell firing?" *Trends Neurosci*, vol. 37, no. 3, pp. 136–45, 2014. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pubmed/24485517>
- [93] J. O'Keefe and N. Burgess, "Dual phase and rate coding in hippocampal place cells: theoretical significance and relationship to entorhinal grid cells," *Hippocampus*, vol. 15, no. 7, pp. 853–66, 2005. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pubmed/16145693>
- [94] E. T. Rolls, S. M. Stringer, and T. Elliot, "Entorhinal cortex grid cells can map to hippocampal place cells by competitive learning," *Network*, vol. 17, no. 4, pp. 447–65, 2006. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pubmed/17162463>
- [95] C. Molter and Y. Yamaguchi, "Impact of temporal coding of presynaptic entorhinal cortex grid cells on the formation of hippocampal place fields," *Neural Netw*, vol. 21, no. 2-3, pp. 303–10, 2008. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pubmed/18242058>
- [96] L. de Almeida, M. Idiart, and J. E. Lisman, "The input-output transformation of the hippocampal granule cells: from grid cells to place fields," *J Neurosci*, vol. 29, no. 23, pp. 7504–12, 2009. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pubmed/19515918>
- [97] S. Cheng and L. M. Frank, "The structure of networks that produce the transformation from grid cells to place cells," *Neuroscience*, vol. 197, pp. 293–306, 2011. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pubmed/21963867>
- [98] S. J. Zhang, J. Ye, C. Miao, A. Tsao, I. Cerniauskas, D. Ledergerber, M. B. Moser, and E. I. Moser, "Optogenetic dissection of entorhinal-hippocampal functional connectivity," *Science*, vol. 340, no. 6128, p. 1232627, 2013. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pubmed/23559255>
- [99] V. H. Brun, S. Leutgeb, H. Q. Wu, R. Schwarcz, M. P. Witter, E. I. Moser, and M. B. Moser, "Impaired spatial representation in CA1 after lesion of direct input from entorhinal cortex," *Neuron*, vol. 57, no. 2, pp. 290–302, 2008. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pubmed/18215625>
- [100] E. J. Henriksen, L. L. Colgin, C. A. Barnes, M. P. Witter, M. B. Moser, and E. I. Moser, "Spatial representation along the proximodistal axis of CA1," *Neuron*, vol. 68, no. 1, pp. 127–37, 2010. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pubmed/20920796>
- [101] T. Van Cauter, B. Poucet, and E. Save, "Unstable CA1 place cell representation in rats with entorhinal cortex lesions," *Eur J Neurosci*, vol. 27, no. 8, pp. 1933–46, 2008. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pubmed/18412614>
- [102] J. Ormond and B. L. McNaughton, "Place field expansion after focal MEC inactivations is consistent with loss of Fourier components and path integrator gain reduction," *Proc Natl Acad Sci U S A*, vol. 112, no. 13, pp. 4116–21, 2015. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pubmed/25733884>
- [103] N. M. van Strien, N. L. Cappaert, and M. P. Witter, "The anatomy of memory: an interactive overview of the parahippocampal-hippocampal network," *Nat Rev Neurosci*, vol. 10, no. 4, pp. 272–82, 2009. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pubmed/19300446>
- [104] T. F. Freund and I. Katona, "Perisomatic inhibition," *Neuron*, vol. 56, no. 1, pp. 33–42, 2007. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pubmed/17920013>
- [105] T. F. Freund and G. Buzsáki, "Interneurons of the hippocampus," *Hippocampus*, vol. 6, no. 4, pp. 347–470, 1996. [Online]. Available: [https://doi.org/10.1002/\(SICI\)1098-1063\(1996\)6:4<347::AID-HIPO1>3.0.CO;2-I](https://doi.org/10.1002/(SICI)1098-1063(1996)6:4<347::AID-HIPO1>3.0.CO;2-I)
- [106] M. M. Jankowski and S. M. O'Mara, "Dynamics of place, boundary and object encoding in rat anterior claustrum," *Front Behav Neurosci*, vol. 9, p. 250, 2015. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pubmed/26557060>
- [107] M. M. Jankowski, J. Passecker, M. N. Islam, S. Vann, J. T. Erichsen, J. P. Aggleton, and S. M. O'Mara, "Evidence for spatially-responsive neurons in the rostral thalamus," *Front Behav Neurosci*, vol. 9, p. 256, 2015. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pubmed/26528150>
- [108] V. Hok, E. Save, P. P. Lenck-Santini, and B. Poucet, "Coding for spatial goals in the prelimbic/infralimbic area of the rat frontal cortex," *Proc Natl Acad Sci U S A*, vol. 102, no. 12, pp. 4602–7, 2005. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pubmed/15761059>
- [109] M. Stemmler, A. Mathis, and A. V. Herz, "Connecting multiple spatial scales to decode the population activity of grid cells," *Sci Adv*, vol. 1, no. 11, p. e1500816, 2015. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pubmed/26824061>
- [110] Q. Tang, A. Burgalossi, C. L. Ebbesen, S. Ray, R. Naumann, H. Schmidt, D. Spicher, and M. Brecht, "Pyramidal and stellate cell specificity of grid and border representations in layer 2 of medial entorhinal cortex," *Neuron*, vol. 84, no. 6, pp. 1191–7, 2014. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pubmed/25482025>
- [111] J. R. Hinman, M. P. Brandon, J. R. Climer, G. W. Chapman, and M. E. Hasselmo, "Multiple Running Speed Signals in Medial Entorhinal Cortex," *Neuron*, vol. 91, no. 3, pp. 666–79, 2016. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pubmed/27427460>

- [112] E. Kropff, J. E. Carmichael, M. B. Moser, and E. I. Moser, "Speed cells in the medial entorhinal cortex," *Nature*, vol. 523, no. 7561, pp. 419–24, 2015. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pubmed/26176924>
- [113] M. J. Milford, G. F. Wyeth, and D. Prasser, "RatSLAM: a hippocampal model for simultaneous localization and mapping," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, vol. 1, 2004, pp. 403–408 Vol.1.
- [114] F. Yu, J. Shang, Y. Hu, and M. Milford, "NeuroSLAM: a brain-inspired SLAM system for 3D environments," *Biological Cybernetics*, vol. 113, no. 5, pp. 515–545, Dec 2019. [Online]. Available: <https://doi.org/10.1007/s00422-019-00806-9>
- [115] J. R. G. Braga, H. F. C. Velho, G. Conte, P. Doherty, and E. H. Shigemori, "An image matching system for autonomous UAV navigation based on neural network," in *2016 14th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 2016, pp. 1–6.
- [116] M. A. Alsheikh, S. Lin, D. Niyato, and H. Tan, "Machine Learning in Wireless Sensor Networks: Algorithms, Strategies, and Applications," *IEEE Communications Surveys Tutorials*, vol. 16, no. 4, pp. 1996–2018, 2014.
- [117] L. Ran, Y. Zhang, Q. Zhang, and T. Yang, "Convolutional neural network-based robot navigation using uncalibrated spherical images," *Sensors*, vol. 17, p. 1341, 06 2017.
- [118] A.-M. Zou, Z.-G. Hou, S.-Y. Fu, and M. Tan, "Neural Networks for Mobile Robot Navigation: A Survey," in *Advances in Neural Networks - ISNN 2006*, J. Wang, Z. Yi, J. M. Zurada, B.-L. Lu, and H. Yin, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 1218–1226.
- [119] M. A. Hannan Bin Azhar and K. R. Dimond, "Design of an FPGA based adaptive neural controller for intelligent robot navigation," in *Proceedings Euromicro Symposium on Digital System Design. Architectures, Methods and Tools*, 2002, pp. 283–290.
- [120] V. A. Torres, B. R. Jaimes, E. S. Ribeiro, M. T. Braga, E. H. Shigemori, H. F. Velho, L. C. Torres, and A. P. Braga, "Combined weightless neural network FPGA architecture for deforestation surveillance and visual navigation of UAVs," *Engineering Applications of Artificial Intelligence*, vol. 87, p. 103227, 2020. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S095219761930212X>
- [121] K. Ren, T. Zheng, Z. Qin, and X. Liu, "Adversarial Attacks and Defenses in Deep Learning," *Engineering*, vol. 6, no. 3, pp. 346 – 360, 2020. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S209580991930503X>
- [122] R. Caruana, S. Lawrence, and L. Giles, "Overfitting in Neural Nets: Backpropagation, Conjugate Gradient, and Early Stopping," in *Proceedings of the 13th International Conference on Neural Information Processing Systems*, ser. NIPS'00. Cambridge, MA, USA: MIT Press, 2000, p. 381–387.
- [123] R. M. Wang, C. S. Thakur, and A. van Schaik, "An FPGA-Based Massively Parallel Neuromorphic Cortex Simulator," *Frontiers in Neuroscience*, vol. 12, p. 213, 2018. [Online]. Available: <https://www.frontiersin.org/article/10.3389/fnins.2018.00213>
- [124] C. S. Thakur, J. L. Molin, G. Cauwenberghs, G. Indiveri, K. Kumar, N. Qiao, J. Schemmel, R. Wang, E. Chicca, J. Olson Hasler, J.-s. Seo, S. Yu, Y. Cao, A. van Schaik, and R. Etienne-Cummings, "Large-scale neuromorphic spiking array processors: A quest to mimic the brain," *Frontiers in Neuroscience*, vol. 12, p. 891, 2018. [Online]. Available: <https://www.frontiersin.org/article/10.3389/fnins.2018.00891>