Department of Informatics
Technical University of Munich

Master's Thesis in Robotics, Cognition, Intelligence

Biologically Inspired Spatial Navigation Based on Head Direction Cells, Grid Cells, Place Cells and Border Cells

Biologische inspirierte räumliche Navigation basierend auf Kopf Orientierungszellen, Gitterzellen, Platzzellen und Grenzzellen

Supervisor Prof. Dr.-Ing. habil.Alois C.Knoll

Advisor Dr.rer.nat. Zhenshan Bing

Author Zhanlun He

Date March 16, 2023 in Garching

Disclaimer

I confirm that this Master's Thesis is my own work and I have documented all sources and material used.

Garching, March 16, 2023

(Zhanlun He)

Abstract

The rodent's ability to navigate in complex environments and find food or revisit previous locations, even in the absence of light, has been extensively studied. Based on behavioral data, several cell types have been identified in the rat's brain, including Head Direction Cells (HDCs) that increase their firing rate based on the animal's head direction, independent of its position, Grid Cells (GCs) in the Medial Entorhinal Cortex (MEC) that fire periodically at multiple locations in the environment and form a triangular grid pattern, Place Cells (PCs) in the hippocampus that fire exclusively at a specific location, and Boundary Cells (BCs) that increase their firing rate when the rat moves close to obstacles in a preferred direction. The prefrontal cortex is believed to record environmental information to construct a cognitive map that can be used for topology-based navigation.
Previous studies have presented individual models for each of these cell types, but there is no general system that combines them all. In this work, we propose a biologically inspired navigation system that includes all these models, enabling interactive functioning between them. The navigation system employs a combination of vector-based and topology-based navigation, which enables the agent to switch between the two modes based on the distance from obstacles. To evaluate the performance of our system, we use a virtual environment to simulate exploration and navigation in different configurations. Our system successfully navigates to the goal with high reliability. Our work provides a possible system for understanding the interaction between the previously identified cells and how a cognitive map is constructed and used for navigation.

Zusammenfassung

Um in einer komplexen Umgebung Nahrung zu finden oder einen früheren Ort zu besuchen, sind Nagetiere in der Lage, ein Ziel zu finden, selbst bei Dunkelheit. Nach der Analyse des Verhaltens von Ratten wurden im Rattengehirn verschiedene Zelltypen identifiziert. Kopforientierungszellen, die im kortikalen und subkortikalen Bereich des Rattenhirns vorkommen, erhöhen die Aktivität nur in einer bestimmten Orientierung unabhängig von der Position des Tieres. Gitterzellen im Medialen Entorhinalen Cortex (MEC) feuern periodisch an mehreren Orten in der Umgebung und bilden regelmäßige Gittermuster, während Platzzellen im Hippocampus nur an bestimmten Orten der Umgebung feuern. Grenzzellen erhöhen die Aktivität, wenn sich die Ratte in bevorzugter Richtung Hindernissen nähert. Der präfrontale Kortex soll Informationen über die Umgebung aufzeichnen, um eine kognitive Karte zu erstellen, die für eine topologiebasierte Navigation verwendet werden kann. In bisherigen Arbeiten der TUM wurden verschiedene Modelle für diese Zelltypen präsentiert und in einer virtuellen Umgebung getestet. Diese Modelle waren jedoch unabhängig voneinander und es fehlte ein allgemeines Modell, das alle Zelltypen integriert. Die Zelltypen interagieren bei der Navigation miteinander. Auf Basis dieser Erkenntnisse schlagen wir ein biologisch inspiriertes System vor, das alle Zelltypen für die Navigation integriert. Wir verwenden eine Kombination aus vektorbasierter und topologiebasierter Navigation, damit der Agent je nach Entfernung zu Hindernissen zwischen ihnen wechseln kann. Zur Leistungsbewertung nutzen wir eine virtuelle Umgebung, um Exploration und Navigation unter verschiedenen Bedingungen durchzuführen. Der Agent erreicht das Ziel zuverlässig. Unsere Arbeit stellt ein mögliches Modell dar, um die Interaktivität zwischen den zuvor identifizierten Zelltypen und die Bildung und Verwendung einer kognitiven Karte für die Navigation zu verstehen.

Contents

1 Introduction **1**

2 Background **5**
2.1 Biological Spatial Cognition 5
2.2 Biological Properties of Cells 5
2.2.1 Grid Cells 5
2.2.2 Place Cells 5
2.2.3 Head Direction Cells 7
2.2.4 Boundary Cells 7

3 Related Work **9**
3.1 Spatial Reference Frames 9
3.1.1 Egocentric Frame 9
3.1.2 Allocentric Frame 10
3.2 Head Direction Model 10
3.2.1 Basic Head Direction Model 10
3.2.2 Head Direction Calibration Model 10
3.3 Boundary Cell Model 12
3.4 Grid Cell Model 13
3.5 Place Cell Model 14
3.6 Navigation 15
3.6.1 Vector-based Navigation 15
3.6.2 Topology-based Navigation 16
3.7 Combinational Navigation 17

4 Problem Statement **19**

5 Methodology **21**
5.1 Overview of Components 21
5.2 Head Direction Calibration Model 22
5.2.1 Model Components 23
5.2.2 Connection Weights 30
5.2.3 Initialization 33
5.2.4 Apply Stimulus and Decode Directions 35
5.3 Boundary Cell Model 35
5.4 Grid Cell Model 36
5.4.1 Single Grid Cell Module 36
5.4.2 Combining Grid Cell Module 37
5.5 Place Cell Model 37
5.6 Cognitive Map 38
5.6.1 Recency Cells 38

5.6.2 Topology Cells . . . 38
5.6.3 Reward Cells . . . 39
5.7 Exploration . . . 39
5.8 Navigation . . . 39
5.8.1 Navigation Phase Controller . . . 40
5.8.2 Movement of Agent . . . 44
5.8.3 HDC Model . . . 44

6 Experiments **45**
6.1 Setup . . . 45
6.2 Grid Cell model . . . 45
6.3 Cognitive Map . . . 47
6.4 Head Direction and Cue Direction Map . . . 47
6.5 Receptive Fields of Boundary Cell Model . . . 49
6.6 Navigation . . . 50
6.6.1 Goal and Sub-goal Localization . . . 50
6.6.2 Navigation in Different Maze Configurations . . . 50

7 Discussion **55**
7.1 Advantage of the Proposed System . . . 55
7.2 Limits of the Proposed System . . . 55
7.2.1 Computer Requirements . . . 55
7.2.2 Straight Topology-based Navigation . . . 56
7.3 Biological Plausibility . . . 56

8 Conclusion and Outlook **57**
8.1 Conclusion . . . 57
8.2 Outlook . . . 58
8.2.1 Additional Cell . . . 58
8.2.2 Improvement of Grid Cell Model . . . 58
8.2.3 Multiple landmarks . . . 58

A Appendix 1 **59**
A.1 Implementation Details . . . 59
A.1.1 Configuration of Computer Environment . . . 59
A.1.2 Hyperparameters of the Models . . . 59

Bibliography **67**

Chapter 1

Introduction

The rat has the ability to locate a goal or revisit a previous location in a complex environment, even in the absence of light. Numerous experiments have been conducted to investigate the correlation between brain cells and spatial orientation. In 1971, O'Keefe and Dostrovsky discovered the first group of neurons associated with spatial navigation in the hippocampus, known as Place Cells (PC), which only fire when the rat is in a specific location in the environment [OKe71; OKe76]. Subsequently, more navigation-related cells have been identified, with the most extensively studied cells being:

- Place Cell (PC):A neuron that exclusively fires at a specific location and signals the animal's position in the environment is known as a Place Cell (PC) [OKe76].
- Grid Cell (GC):Grid Cells (GC) are a type of neuron that fires periodically at multiple locations throughout the environment, forming a regular triangular pattern that is independent of the animal's speed or orientation [Mos08].
- Head Direction Cell(HDC): Head Direction Cells (HDC) are a type of neuron found in various regions of the brains of mammals, such as rats and mice, that signal the animal's head orientation irrespective of its position in the environment [Tau90].
- Boundary Cell (BC): Boundary Cells (BC) are a type of neuron that fire at a specific distance and orientation when the animal encounters obstacles in the environment [Lev09].
- Speed Cell: Speed Cells are a type of neuron found in the entorhinal cortex that fire in a linear relationship with the running speed of animals [Kro15].

Multiple models based on Spiking Neuron Networks (SNN) have been proposed for Simultaneous Localization and Mapping (SLAM) to enable agents to navigate.
The GC model is comprised of multiple GC modules of varying scales, which can be implemented using either a nested or combinatorial approach. In the nested approach, the GC module with the largest scale covers the environment size, while the smallest determines the model's resolution [Edv17b]. It has been proposed that the GC model can be constructed using GC modules with scales that increase linearly by a factor of 1.42 [Ste+12]. On the other hand, the combinatorial approach, introduced by Bush [Bus15], combines multiple GC modules with smaller scales to create a compact combinatorial code that is unique across a distance determined by the common multiple of the GC scales [Edv17b] [Eng21]. These models have been proposed for simultaneous localization and mapping (SLAM), enabling agents to navigate in various environments.
It was proposed that weak spatial inputs are strengthened through mechanisms in the hippocampus circuit, resulting in the firing of PCs [Edv17a]. Another proposal suggests that the

input to PCs is the weighted output of GCs with varying scales [Mcn+06].
The model of HDC utilizes two feedback models, namely the simple feedback model and the place encoding feedback model. Skaggs proposed a model for HDC that utilizes Continuous Attractor Neural Network (CANN), which consists of a single ring of neurons with symmetric connections to provide HD information [Ska+94]. K. Zhang later extended this model by adding an Egocentric Cue Direction (ECD) ring, which provides cue information to the HD ring via Hebbian-learning connections to all neurons in the CANN. The ECD ring responds to cue information in an egocentric frame [Zha96].
In the place encoding feedback model, the agent's allocentric position is utilized to minimize HD errors. Two similar but different models are the ones proposed by Bicanski and Burgess [BB18] and Arleo and Gerstner [AG00], which have the same inputs but different processing components [Nit21]. Another HDC model was presented by Amir [Sew20], which consists of a basic HDC ring and two shift layers, each composed of hundred neuron models inspired by Ulises [PB18]. Additional cue information is fed to this model to decrease the error of agent position estimation. This model also utilizes a First Glance Learning (FGL) model to store the agent's position and vector from the agent to the landmark. This information is used to correct the HD error if the agent revisits the place within the range and angle of the cue [Nit21].
Bicanski and Burgess proposed a BC model to convert egocentric information into allocentric representation for supporting imagery and recollection [BB18]. Inspired by this idea and the HDC model by Amir, Camillo presented a model that uses the HDC network to transform egocentric information to allocentric information [Hey21].
The prefrontal cortex is thought to store information about the environment to create a cognitive map that is necessary for topology-based navigation [Eic17]. Erdem and Hasselmo proposed a model with different cell types in the prefrontal cortex, including recency cells, topology cells, and reward cells [EH12]. In Tim's model, the prefrontal cortex cell model is linked to the PC model. The recency cells store information about recently visited locations, the topology cells contain the relationship of neighborhood between PCs, and the reward cells provide information about rewards for goal-driven navigation. This information is updated when the agent enters a new place [Eng21].
The navigation concept can be categorized into vector-based navigation and topology-based navigation, which are used in different situations [Eng21]. Vector-based navigation is used when the agent can move directly towards the goal without any obstacles, while topology-based navigation is used when the agent has to navigate around obstacles to reach the goal [Edv20]. To build a cognitive map for topology-based navigation, a graph-based approach or a network of neurons using prefrontal cortex cells can be used. During navigation, if the goal vector is blocked, the agent will start searching for a non-blocked vector from the goal to a graph node. Alternatively, forward look-ahead can be used, where the agent virtually searches for the path with the highest reward in a predefined number of steps to find a sub-goal [EH12]. Tim proposed a hybrid navigation method that combines vector-based navigation and topology-based navigation, depending on the distance and orientation of the obstacles from the agent [Eng21].
The purpose of this study is to introduce a system that integrates the necessary cell models for navigation, including the PC model, GC model, HDC model, and BC model. The speed cell model is not included due to the limited number of models available and the constant speed of the agent in the simulation. A simple neuron model based on SNN is used to construct the network of basic models. While the details of these biological cells are still unclear and the existence of some cells lacks biological evidence [May15], the proposed system reflects the biological nature of these cells, such as the firing rate increasing as the agent approaches obstacles and decreasing as the agent moves away from them. Though it is challenging to

provide biological evidence for the proposed system, it demonstrates the potential of these models to accurately represent the neural activity underlying navigation.
The proposed system was evaluated in a real-time physics simulation environment called PyBullet, where exploration and navigation tasks were performed under different configurations. The results showed that the agent was able to reliably reach the goal, indicating the effectiveness of the combined model for supporting navigation and spatial cognition tasks.

Chapter 2

Background

2.1 Biological Spatial Cognition

It is widely accepted that information from GC, PC, and HDC forms an internal cognitive map in the brain that supports spatial localization and navigation in animals and humans [Pou93]. A cognitive map is a cognitive representation that serves to acquire, store, and reorganize information about the environment [Tol48]. The hippocampus is believed to be responsible for creating and integrating both spatial and non-spatial information in the brain. O'Keefe first discovered PC in the hippocampus [OKe71], while Moser identified GC in the Entorhinal Cortex (EC) and J. Ranck reported HDCs in the rat dorsal presubiculum [Tau84]. The neurons responsible for detecting boundaries were first predicted by Tom Hartley [Har+00], and border cells have since been discovered in the rat brain [Sol+09a]. Figure 2.1 illustrates the main hippocampal formation.

2.2 Biological Properties of Cells

2.2.1 Grid Cells

Moser discovered GCs in the Medial Entorhinal Cortex (MEC) [Mos08]. These cells were found to fire at multiple fields forming a regular triangular pattern across the entire environment explored by the animal, as shown in Figure 2.2 [Haf+05]. The distance between fields, their orientation, and phase determine each grid, which serves as a spatial coordinate system. Landmarks or well-known locations, such as a home base, can function as an origin to enable the determination of the rat's position in the environment [Tej+18]. This function allows animals to use path integration to track their distance and direction, estimating their position, which is known as neural odometry [Tej+18; Eng21].

2.2.2 Place Cells

In 1971, O'Keefe first reported that cell firing in the hippocampus was related to the current location of a rat [OKe71]. These cells were named Place Cells [OKe76] and they only fire at specific locations in the environment, as shown in Figure 2.2. Unlike GCs, which fire periodically at several locations, a PC only fires within a place field and can be reused in different environments. Because a PC represents the location of a rat in an allocentric frame in an environment, they are believed to form the basis of the cognitive map [Mos08].

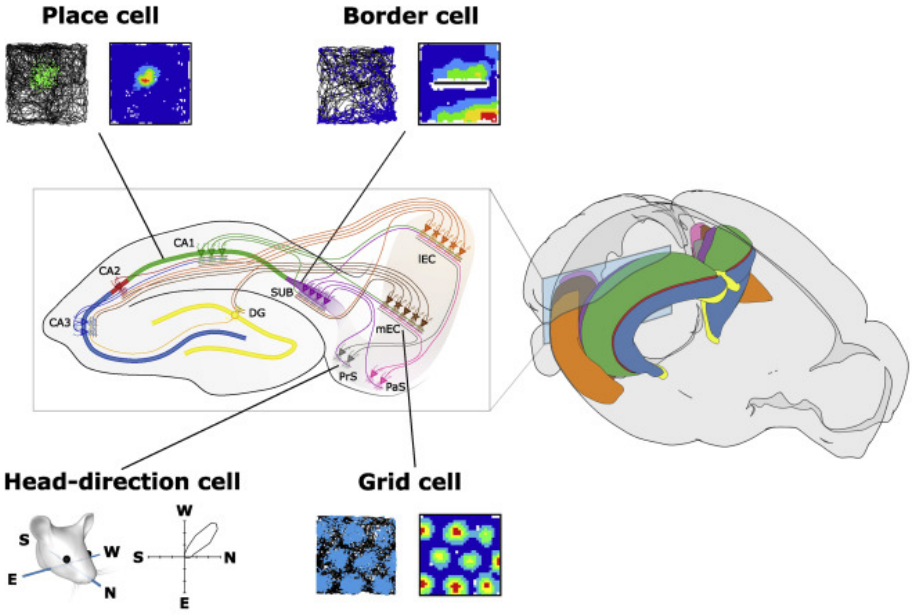

Figure 2.1: Standard description of connectivity and neural representation of space within the rodent hippocampal formation[Nil22]. The rodent hippocampal formation is interconnected through projections from the medial and lateral entorhinal cortex (mEC and lEC) to CA3 and CA1 via the dentate gyrus (DG). CA3 also sends projections to other networks. DG, mEC, and lEC project to CA3, CA1, and subiculum (SUB), with CA1 projecting back to mEC/lEC through SUB, and SUB also projecting directly to mEC/lEC via presubiculum (Prs) and parasubiculum (Pas). This diagram does not include all projections. The illustration of place cells (PCs), boundary cells (BCs), head direction cells (HDCs), and grid cells (GCs) shows a rodent's trajectory with spiking activity depicted as a heatmap, where warmer colors indicate increased activity.

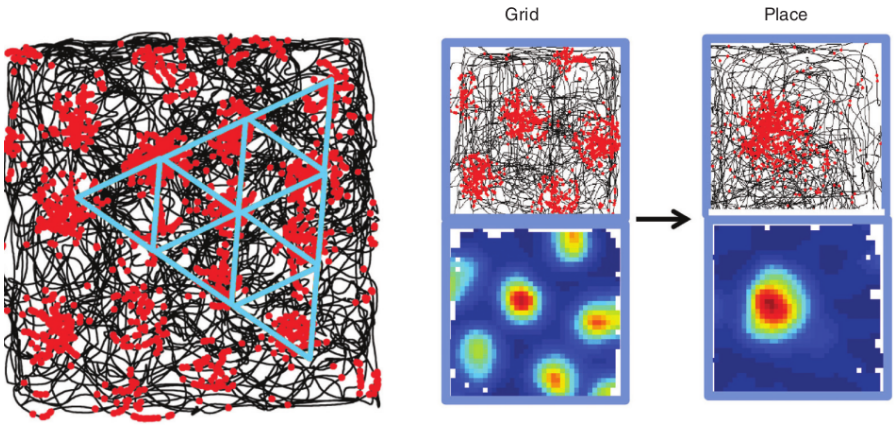

Figure 2.2: Grid cells and place cells. The left image depicts a GC from the entorhinal cortex of a rat brain, showing the black trace of the rat's trajectory and red dots representing spikes in a maze with a 1.5m diameter square. The regular triangles are formed by connecting the distribution of the spikes. The right image shows both GC and PC, with the top panel depicting the spike trajectory and the bottom panel displaying a color-coded rate map where red indicates high activity and blue indicates low activity [May15].

It has been suggested that GCs are the primary input of PCs [Eng21], and that the input to PCs is the Fourier-like sum of weighted GC outputs [Mcn+06]. Another suggestion is that weak spatial input is amplified by mechanisms in the hippocampal circuit and sent to PCs. The properties of PCs have also been found in young rats where GCs are still developing, suggesting that other cells, such as Border Cells (also known as Boundary Cells), are relevant to PC firing [May15].

2.2.3 Head Direction Cells

HDCs were first reported by James Rank in the dorsal presubiculum of rats [Tau84]. They have also been found in other animals such as mice and chinchillas [Yod09]. HDCs are neurons located in several regions of the brain that fire in a horizontal plane associated with an animal's orientation relative to its environment, independent of its location. Their discharges are tuned to a specific direction depending on landmarks and ego-motion cues, similar to how a compass relies on the geomagnetic field [Yod09]. Each HDC can be characterized by a few parameters based on its tuning curve, often plotted on an x-y graph with firing rate on the y-axis and head direction on the x-axis (see Figure 2.3). HDCs also work in darkness without visual cues, relying only on self-movement information, a process known as path integration. In this process, the animal registers its starting location and head direction and then estimates its current location and orientation based on the integration of internal self-motion information, such as vestibular inflow. Angular head velocity (AHV) is believed to be an important input for driving HDC behavior [Tau07]. Cue information, which can be obtained from any sensory modality such as visual, auditory, and olfactory, is also involved in controlling the preferred firing direction [Tau09] and can be used to calibrate the estimated head directions based on HDCs [Tau90][Nit21]. However, calibration using cue information works only after a learned stable mapping between the cue information and head direction has been established by the animal [Mcn+06].

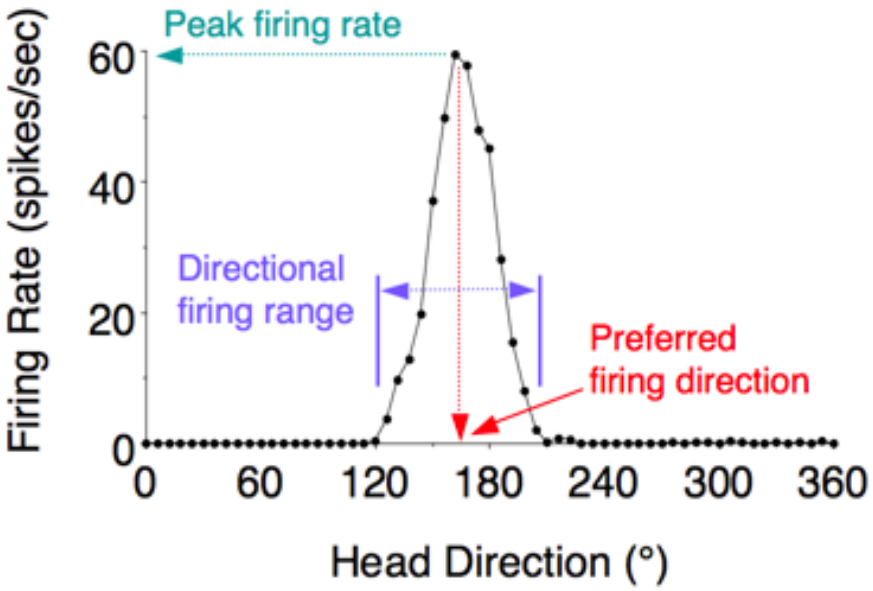

Figure 2.3: Firing rate vs. head direction plot to characterize the firing properties of head direction cells [Tau09].

2.2.4 Boundary Cells

Boundary cells (also known as border cells or boundary vector cells) are neurons which fire only if animals encounter obstacles in specific distance and orientation from animals [Hey21]. The existence of these cells was first predicted based on properties of place cells. It was found by O'Keefe and Burgess that a few place cells fired only at circumscribed area of the environment and tended to fire especially in the location where the shape and size of the environment changed [OB96]. Based on this observation, boundary vector cells (BVCs)

were proposed by Burgess and O'Keefe groups [Har+00]. In later work, the cells with these properties were discovered in several regions of the hippocampal formation: the subiculum [Bar+06], presubiculum [Sol+09b], and entorhinal cortex [Boc+10] and were named variously as "boundary vector cells", "border cells" and "boundary cells". These cells have a spatial receptive field. When an obstacle is within the range of the receptive field and animals get closer to obstacles, the neurons' firing rate increases and the receptive field gets smaller, as shown in Figure 2.4 and Figure 2.5.

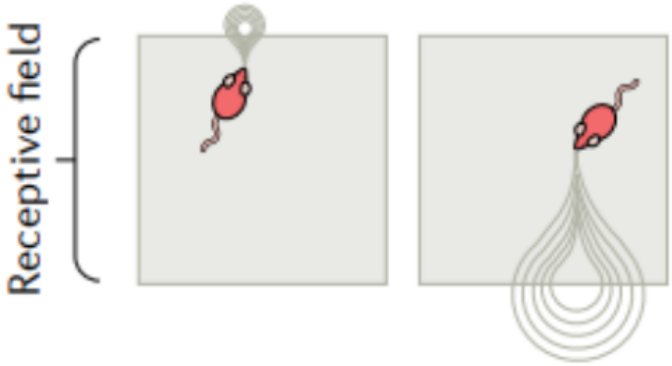

Figure 2.4: Two different receptive field.(Left) When a rat is close to a boundary, the receptive field of the boundary cell is smaller. (Right) when the rat is far away from the obstacle, the receptive field of the boundary cell is larger [BB20] [Hey21].

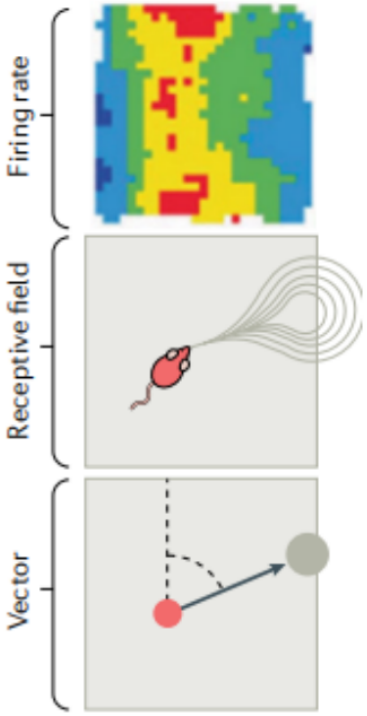

Figure 2.5: (Top) A firing rate map of a boundary vector cell, which shows the cell's firing rate in different locations of the environment. (Middle) An illustration of the receptive field of the cell at a specific distance and orientation from the rat to the obstacle. (Bottom) A vector representing the direction from the rat to the receptive field of the cell. These figures provide a visual representation of how boundary vector cells respond to the presence of boundaries in the environment [Hey21] [BB20].

Chapter 3

Related Work

In this chapter, we will provide a brief overview of the models that have been proposed for place cells (PCs), grid cells (GCs), head direction cells (HDCs), and boundary cells (BCs).

3.1 Spatial Reference Frames

Tolman's idea of a cognitive map was introduced to represent the spatial information and navigation of animals in a known environment [Tol48]. The traditional distinction between two types of frames for representing spatial information is the allocentric frame and egocentric frame, which are illustrated using an example in Figure 3.1.

3.1.1 Egocentric Frame

The egocentric frame is a self-centered representation where the agent's head is the origin of the coordinate system. It is a type of subject-to-object relation [Col+17] and is dependent on the agent's position and point of view. This frame is depicted in orange in Figure 3.1. It is dependent on the agent's movement in translation and rotation. Thus, the location of a cue fixed in the allocentric frame changes in the egocentric frame if the agent changes its position or orientation [Nit21].

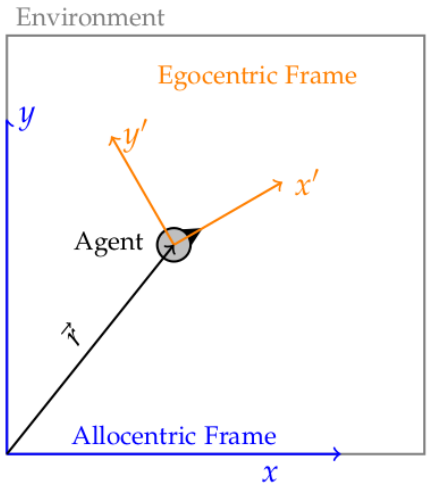

Figure 3.1: Reference frames: the egocentric frame in orange is body-centered representation, the allocentric in blue is world-centered representation [Nit21].

3.1.2 Allocentric Frame

The allocentric frame represents spatial information based on the position of objects in a world-based coordinate system and the relationship between objects [Col+17]. It is depicted in blue in figure 3.1 and is tied to the bottom left corner of the environment, although the origin in this frame could also be any other meaningful position, such as a landmark. The location of an animal in this frame is independent of the agent's point of view.

3.2 Head Direction Model

Head Direction Models can be divided into two categories: basic HDC models and HDC calibration models. In the basic HDC model, cue information is not involved in estimating the head direction. The estimation of the head direction is based on path integration, which can lead to the accumulation of errors over time. On the other hand, the HDC calibration model uses cue information to correct the errors.

3.2.1 Basic Head Direction Model

A theoretical model was presented by Zhang in 1996, proposing a basic dynamic model based on Gaussian-like directional tuning curves and shift mechanisms [Zha96]. Based on this idea, Amir introduced a basic HDC model consisting of an HDC Attractor Network and two shift layers (one for left and one for right) [Sew20]. The HD Attractor consists of 100 spiking neurons that mimic the biological neural network, with each neuron corresponding to a specific orientation of the agent known as the Preferred Firing Direction (PFD). The activity network is based on the activity of each neuron in its PFD, as shown in the left part of Figure 3.2. These neurons cover the entire circle equally and connect with each other. The neurons connect with close neurons through excitatory synapses and with far neurons through inhibitory synapses, as shown in the right part of Figure 3.2. With local excitation and global inhibition, using carefully calculated synapse weights, a stable activity peak can be maintained after initialization. The stable activity peak can be shifted by the shift layers, which use angular velocity as input. Each shift layer also has 100 neurons to cover the entire circle and get the same PFD as the HD Attractor. Each neuron in the shift layers is connected to all neurons of the HD Attractor with half the weights as the weights between neurons in the HD Attractor. The connection weights of both shift layers are mirrored so that they cancel out if there is no external stimulus. Each neuron in the shift layer has an excitatory connection to the HDC in its shift direction and inhibitory connections in the other direction. In this model, the agent can use angular velocity as input to perform path integration, which leads to the accumulation of error in estimation. To correct this error, researchers have proposed several models using visual cues.

3.2.2 Head Direction Calibration Model

According to Dominik et al. [Nit21], there are two main types of calibration models: the simple feedback model and the place encoding model. The simple feedback model relies solely on egocentric cue direction as input to infer HD, whereas the place encoding model additionally incorporates information about the agent's position.

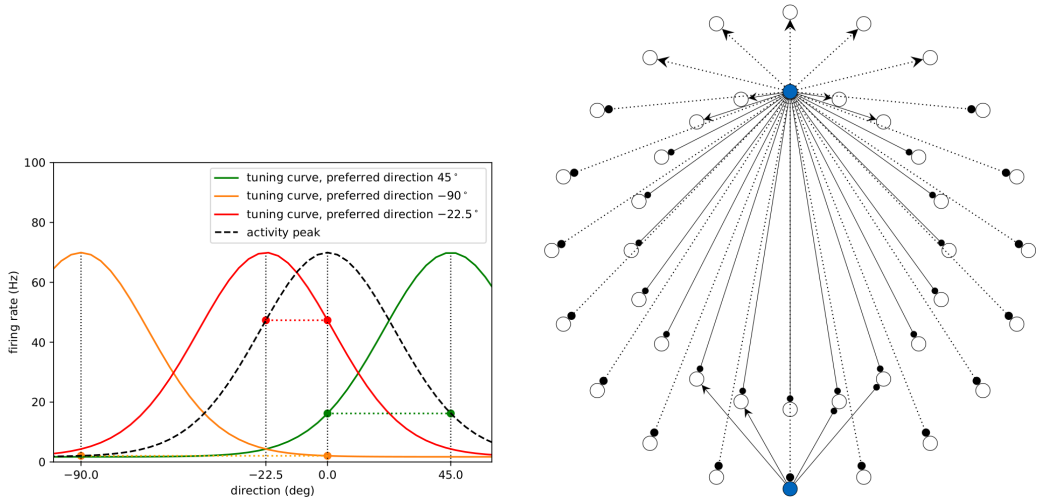

Figure 3.2: Schematic of HDC Attractor.(Left) Three curves of neurons with different Preferred Firing Direction(PFD) (-90°,-22.5° and 45°) in the HD network,the HD is 90°. The activity profile of the network is shown as a black dashed line, which maps the PFD of the HDC network. The activity profile looks similar to the tuning curve of an individual neuron with PFD equal to HD [Sew20]. (Right) Illustration of HDC Attractor and shift layer. The figure depicts an HDC Attractor Network at the inner circle, and a shift right layer at the outer circle (the shift left layer is not shown as it is a mirror of the shift right layer). Each neuron in the shift layer is connected to all neurons in the HDC network. A stable peak in activity is achieved through local excitation (arrowhead) and global inhibition (solid circles) by carefully chosen weights. An external stimulus applied to the shift layer can shift the activity profile in the HDC network [Sew20].

Simple Feedback Model

There are two calibration models for Head Direction (HD) cells: the Simple Feedback Model and the Place Encoding Model, as proposed by Dominik et al. [Nit21]. In the Simple Feedback Model, the Egocentric Cue Direction (ECD) ring for perceiving cues is directly connected to the HD ring. This model was first presented by K. Zhang [Zha96]. It consists of two HD rings that can be imagined as two identical rings stacked on top of each other. All neurons in the rings are interconnected. An external asymmetric stimulus to any neuron of the rings can cause an activity profile in the shape of a Gaussian curve in the same position of both rings, allowing the orientation of the agent to be decoded. One of the rings has excitatory connections to the clockwise neighbored neurons and inhibitory connections to the counter-clockwise neighbored neurons, while the other ring is in the opposite way. So, the two rings try to shift the activity profile in opposite directions, and if the magnitudes of the activity profile in both rings are the same, the shift stimulus cancels out. During an external stimulus, the ring for that direction gets excited and is able to shift the activity profile in both HD rings. The neurons in the visual detector ring have a one-to-one connection to the first HD ring, and the connections are calculated by one-shot Hebbian learning. When the cue information is perceived by the visual detector ring, the neurons are activated to control the position of the activity profile in both HD rings. However, the problem with this model is that the cue must be infinite or the agent's movement must be in a straight line originating from the cue position. If the agent moves around the cue or if the visual cue is unstable, the offset between the HD ring and the visual detector becomes fixed under an infinite distal cue, rendering this model useless. Another model by Skaggs [Ska+94] consists of a Continuous Attractor Neural Network (CANN) and a visual feature detector. The CANN has a similar function to the HD ring in the model by K. Zhang [Zha96] but has only one ring with neurons connected with each other for path integration. Each neuron in the visual detector ring connects through Hebbian learning to all neurons of the CANN instead of one-to-one connection in the model

by K. Zhang. As long as the cue is in a constant angular consistent with the indicated HD of the agent, the connections are strengthened. This model can evaluate the stability of the cue but is not suitable for proximal cues, as the cue is treated as unstable if the agent moves around it.

Place Encoding Model

The model proposed by Bicanski and Burgess [BB16] incorporates a calculation of the offset between visual cues and HD at different positions, which is used to correct HD. By associating HD with ECD in discrete places, this model can use both distal and proximal cues, and unstable cues can be disconnected from HD. Arleo and Gerstner [AG00] presented a similar model in which cue information in an egocentric frame is used to reset HD, with the accuracy of the decoded position needing to be above a given threshold and the time between calibration steps under a certain limit.
Milford et al. [MWP04] proposed a model consisting of two three-dimensional attractor cell networks: the Pose Cell Network and the Local View Cell Network. The Pose Cell Network encodes position in the first two dimensions and orientation in the third dimension for path integration, while the Local View Cell Network encodes the color, distance, and bearing of cues in egocentric frames. Connections between neurons in both networks are strengthened through Hebbian learning if both networks are active at the same time, and weakened if both networks are active at different times. However, the lack of biological plausibility in the cells used in this model has been noted [Nit21].
Dominik's model combines the basic HD model by Amir [Sew20] with a calibration circuit [Nit21]. The basic HD model is the same as the model by Amir, while the calibration circuit includes a visual detector, ECD ring, Allocentric Cue Direction (ACD) ring, position encoder, Adder, and Subtractor. When a cue is detected by the visual detector, ACD information can be derived by HD and ECD via the Adder, and HD information can be inferred based on ACD and ECD via the Subtractor. The position encoder stores ACD information when the agent enters the range of the cue for the first time, and resets HD based on the stored information if the agent enters the range of the cue again.

Cue Weighting model

The model proposed by Page et al. [Pag+14] allows for the use of multiple cues with different weights dependent on their reliability. The model consists of a CANN-like Skaggs' model for HD and a visual ring for ECD. Each neuron in the visual ring is connected to all neurons of the CANN, with one-to-one connections at the beginning. In the event of a conflict between HD and ECD, the activity in the visual ring will shift the activity profile in HD, and the weight of connections will be recalculated in such a way that the HD activity packet moves to the position indicated by the visual ring. This enables the calculation of the HD angle based on the average angle indicated by the visual cue and the idiothetic cue.
The model by Jeffery et al. [JPS16] combines the properties of the Skaggs and Page models. In the event of conflicts arising from two different cues, an average position is calculated to reset the position of the HD activity packet. Additionally, the HD ring can be disconnected for unstable cues.

3.3 Boundary Cell Model

Bicanski and Burgess proposed a theoretical model to explain how egocentric information interfaces with HD in the brain to enable spatial cognition [BB18]. The model aims to trans-

form egocentric representations into allocentric representations for storage, supporting imagery and memory.
A computational model based on their theoretical framework was presented by Camillo [Hey21]. This model uses 51 sensors to detect boundaries for calculating egocentric activity. A transform circuit, consisting of 20 layers, is used to transform the egocentric representation into an allocentric representation.
The 20 layers are equally distributed around a circle, with each layer having a preferred HD. Via connections from these layers, the layer that matches the current HD is activated, while other layers are inhibited. After normalization of the sum of these layers, the representation in the allocentric frame can be calculated.

3.4 Grid Cell Model

The concept of a coordinate system in rodents is demonstrated using a one-dimensional path. As the rodent moves along a linear path, grid cells (GCs) fire at regularly spaced locations, determined by the grid scale and starting position. This firing pattern can be represented by the equation $x = x_0 + k * \lambda$, where $k \in N$ and x_0 is the location of the first GC spike, as shown in the top section of Figure 3.3.
The linear path is divided into grid-scale units, and multiple grid cells (GCs) are combined to

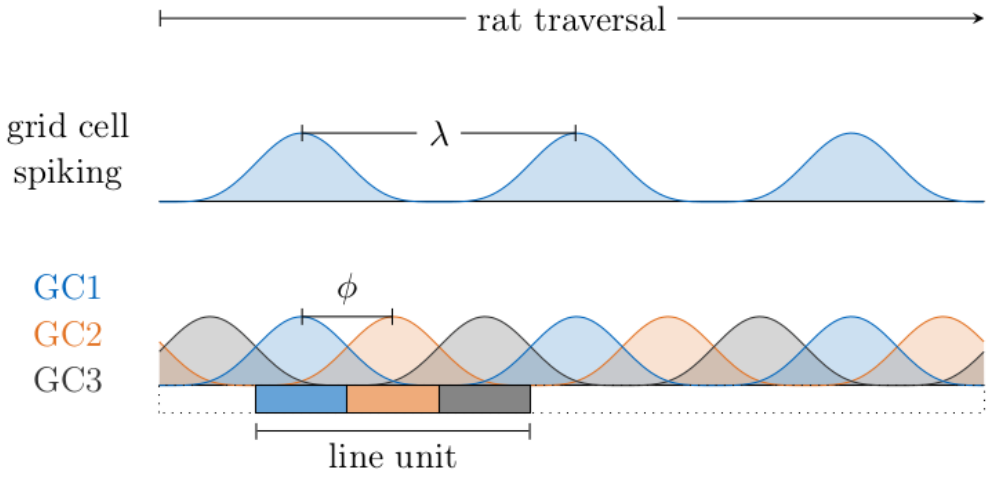

Figure 3.3: Discretization of linear path using combination of modules with a phase offset.(Top) A single GC model with a scale of λ is shown to fire periodically at several locations. (Bottom) Combination of multiple GC modules with the same scale but different phase offsets, denoted by ϕ.

ensure a firing always occurs. These GCs have the same scale but different phase offsets of ϕ between them. The location of the rodent can be described as $x = x_0 + n * \phi + k * \lambda$, where n represents the firing of the n-th GC and λ is the wavelength of the GC firing pattern. See the bottom part of Figure 3.3 for a visual representation. Using this equation, the rodent's relative position within a line unit can be determined [Eng21].
In a 2D environment, the spikes of a GC form a regular triangular pattern, and several GC modules with the same scale and orientation but different phase offsets are used to establish a coordinate system. However, due to the periodic nature of GC firing, the rodent does not know which unit it is in. To improve the accuracy of self-location, two approaches are used: the nested approach and the combinatorial approach.
The nested approach, proposed by Edvardsen [Edv17b], utilizes multiple GC modules with

different scales, where the largest scale covers the entire environment and the smallest scale determines the grid resolution. The scale of these GC modules should increase by a factor of 1.42, as suggested by Stensola et al. [Ste+12], with a grid scale of 1m for linear extrapolation or 10m for exponential extrapolation, as proposed by Hafting et al. [Haf+05]. Tim [Eng21] presented a GC model based on this approach, which consists of six modules with neurons having preferred direction in two axes that form regular triangular patterns. This model's spike is updated based on the velocity vector of the agent. Although the nested approach lacks biological evidence, it is easy to implement and robust to errors.
On the other hand, the combinatorial approach combines multiple GC modules with different scales, where the smallest common multiple of all GC scales characterizes it [Edv17b]. Adding an extra GC module can exponentially increase the covered environment size. This approach is highly biologically plausible, but it can lead to significant errors due to the imprecision of a GC.

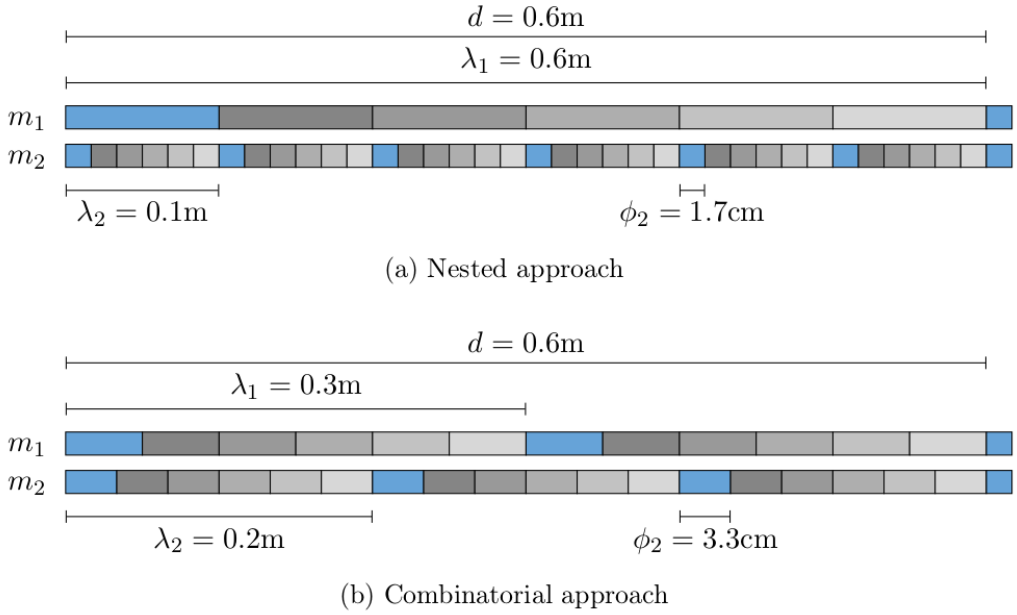

Figure 3.4: Schematic of HDC Attractor.(Left) Three curves of neurons with different Preferred Firing Direction(PFD) (-90°,-22.5° and 45°) in the HD network,the HD is 90°.

3.5 Place Cell Model

The grid cells (GCs) are believed to provide input to the place cells (PCs) [Eng21]. McNaughton et al. suggested a model where the connections from the GCs to the PCs are represented by a Fourier-like sum of weighted connections with different scales [Mcn+06]. In Zhou's model [Zho21], the GCs are connected to the PCs via Hebbian learning. Hebbian learning is a theory that explains synaptic plasticity in neurons and states that the synaptic efficacy will increase if a presynaptic cell repeatedly and continuously stimulates a postsynaptic cell. This theory is often summarized as "cells that fire together wire together" [LS92]. Tim proposed a simple model where the connections from the GCs to the PCs have a weight of one if the spiking value of the GC is positive [Eng21]. The average weighted and normalized spiking value of the GCs is then the spiking value of the PC. A new PC is generated if this value is below a given threshold. Each place cell is linked to a set of three prefrontal cortex cells to build a cognitive map.

3.6 Navigation

There are three main categories of navigation approaches: vector-based, topology-based, and a combination of both [Eng21].In vector-based navigation, the animal uses information about its movement direction and speed to calculate its position relative to a starting point. This approach is useful when navigating in open environments where landmarks are sparse. Topology-based navigation, on the other hand, relies on the animal's ability to recognize and remember the spatial relationships between different landmarks, creating a mental map of the environment. This approach is particularly useful in complex environments with many landmarks, such as mazes or natural environments. Finally, a combination of both approaches can be used, where vector-based navigation is used to estimate the animal's position in open spaces and topology-based navigation is used to navigate in more complex environments.

3.6.1 Vector-based Navigation

During vector-based navigation, the agent navigates by following a calculated vector between its current and goal positions. This approach is suitable for obstacle-free environments, as the GCs can decode the agent's position accurately. By comparing the GC firing at the current position and the goal position, a goal vector for navigation can be determined. Two mechanisms are used for performing this comparison: directed coding and directed search.
In directed coding, an additional network is used to decode the vector between the GC firing at the current position and the goal. Several models using directed coding have been proposed by researchers. For example, the Distance Cell model by Fiete et al. [FBB08] uses the GC phases across modules to decode the absolute current and goal location. Distance cells are neurons that encode a unique location along a directional axis, and an array of such neurons can be used to compute linear displacement. Another alternative is the Rate-coded vector cell model, in which an array of vector cells encodes specific displacements from the current position along a single directional axis. Linear displacement can be decoded directly from the phase difference between GC representations at the current and goal locations across modules.
Edvardsen [Edv15] proposed a model that utilizes phase-offset detectors consisting of neurons for decoding. The first input of the phase-offset detector is the weighted vector s_m of GC activity, which is calculated based on the agent's speed and direction. The second input is the weighted vector t_m of GC activity. The phase-offset detectors aim to find the required direction to build for the offset in the GC pattern between s_m and t_m. Each phase-offset detector j is associated with an origin location x_j and preferred direction θ_j. The neuron is tuned so that there is also an activity peak near the location $z_j = x_j + \delta * \theta_j$ in the GC-encoded target location if an activity peak arises near the origin location x_j in the neuron sheet of path integrator in GC modules. Here, δ is the predefined length of the offset. In other words, if an offset of length δ in direction θ_j between activity patterns in s_m and t_m arises, the detector responds strongly, as shown in Figure 3.5. A set of motor output neurons is connected to the phase-offset detectors to obtain a motor direction signal, giving the direction towards the goal, and a motor strength signal, checking if the agent has arrived at the goal location or should keep going. However, the disadvantage of this model is that a sufficient number of phase-offset detectors with various origin locations x_j and different preferred directions θ_j are required [Edv15].
Another method to determine the vector between the current and goal locations is through a direct search along specific axes, starting from either location. This approach aims to calculate the relative position and utilizes simulated movement signals, which are decoupled

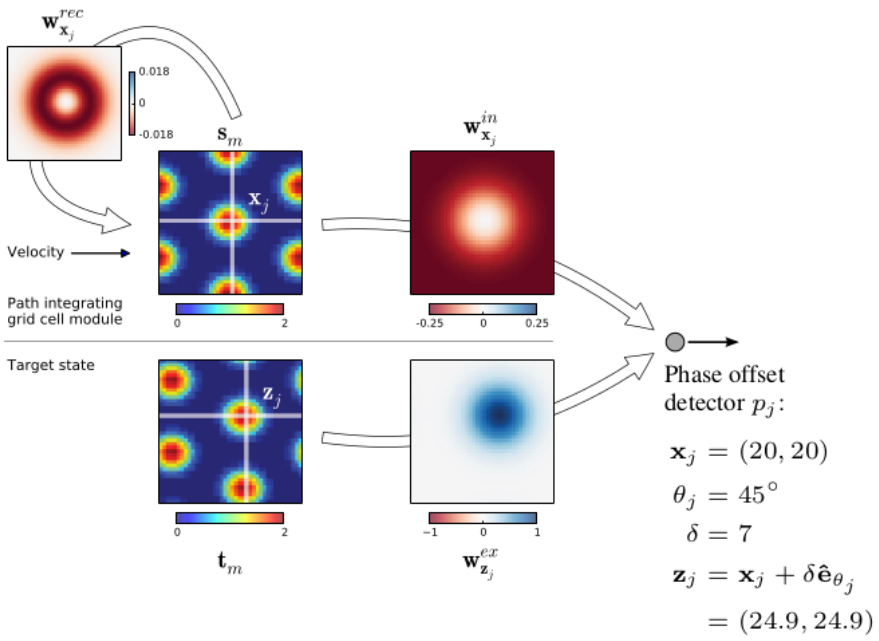

Figure 3.5: Example of a phase offset detector p_j. The 40 x 40 matrices depict the weighted vectors of GC activity in the current location (s_m) and the desired target location (t_m), which are used as inputs for the phase-offset detectors [Edv15].

from the animal's actual motion, to update the GC spatial representation. One example of this is the linear look ahead method proposed by Erdem and Hasselmo, where the agent begins simulated movement from its current location along multiple equally spaced directions, with more directions leading to a higher likelihood of finding the goal [EH12]. Another approach is the two-axis linear search proposed by Bush et al., where the search is performed only along two linearly independent axes. The goal vector is then computed using a linear combination of the projections of the spiking vector on these two axes [Bus15].

3.6.2 Topology-based Navigation

Topology-based navigation is a method used when an agent encounters obstacles and must find an alternate path. The process involves two main steps: storing information to construct a cognitive map during exploration, and using this map for navigation. Edvardsen proposed a graph-based approach for building a cognitive map [Edv20], where a graph node corresponding to a place cell is created when the agent enters a new location. A snapshot of the current grid cell (GC) activity is saved and linked to the graph node. Graph nodes for two neighboring locations are also linked. The goal node is marked, and after extensive exploration, more nodes are added to the graph to construct a detailed cognitive map. Erdem and Hasselmo [EH12] proposed a similar approach, but using prefrontal cortex cells instead of graph nodes. When an agent enters a new location, it creates a prefrontal cortex to store the place information and reward gradient towards the goal. Building upon this, Tim suggested a cognitive map consisting of three prefrontal cortex cells linked to every place cell [Eng21]. Each set includes a recency cell, a topology cell, and a reward cell. The recency cell stores the agent's visited history, while the topology cells provide a matrix for the relationships between neighbors. The topology cells save the reward gradient towards the current goal.

3.7 Combinational Navigation

The combination of the two approaches discussed above has been proposed for navigation [Edv20]. The agent initially employs vector-based navigation as its default mode. If an obstacle blocks the path, the agent switches to topology-based navigation. When the path becomes unobstructed again, the agent switches back to vector-based navigation. A cognitive map built using graph-nodes [Edv20] or prefrontal cortex cells [Eng21] can be utilized in this combined navigation strategy.

Chapter 4

Problem Statement

The present work aims to create a system that combines various computational models for Head Direction Cell, Place Cell, Grid Cell and Boundary Cell in order to mimic animal navigation behavior. These basic models include the Head Direction Model by Dominik [Nit21], the Grid Cell and Place Cell models, the navigation approach by Tim [Eng21], and the Boundary Cell Model by Camillo [Hey21]. The goal is to design a system that is compatible with all basic models, with any parameter modification in one model not causing issues in others. The system should perform similar functions, such as obstacle detection, only once to optimize efficiency.
The proposed system should provide information about the agent's head direction and firing map when encountering obstacles. The agent should be able to store information for a cognitive map during navigation and use it to reach the goal reliably, switching between vector-based and topology-based navigation as needed. The environment should include a single cue, multiple obstacles, and a single goal, with the agent navigating through different configurations of the environment.
Overall, the aim of this work is to create a system that combines basic models to provide information about orientation, obstacle detection, and cognitive map creation for reliable and efficient navigational performance.

Chapter 5

Methodology

In this chapter, we will present the proposed system, which combines the Head Direction Calibration model [Sew20][Nit21], Boundary Cell model [Hey21], Grid Cell model, Place Cell model, and navigation system [Eng21]. We will begin by providing an overview of the system and then proceed to explain the individual basic models in detail. Finally, we will provide information about exploration and navigation within the system.

5.1 Overview of Components

The fundamental components of the system are the neurological basic models, which are interconnected as illustrated in Figure 5.1.

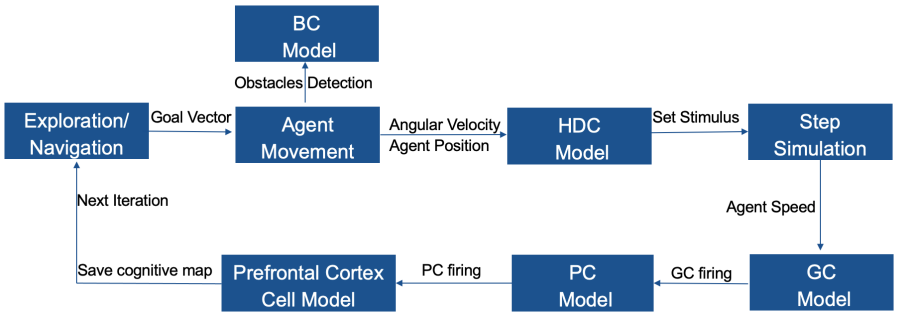

Figure 5.1: Overview of the system. The system utilizes the goal vector to perform obstacle detection, which in turn is used to update the Boundary Cell firing map. The agent's position and angular velocity are used to decode the head direction information. The firing of Place Cells can be computed based on the speed of the agent and the firing of Grid Cells. The relevant information for building the cognitive map is stored in the linked prefrontal cortex cells. This process is repeated in each iteration of exploration or navigation until the agent reaches the goal or the cognitive map is complete.

During the first step of the system, the agent identifies the goal vector by performing either exploration (by searching for subgoals using a predefined list of locations) or navigation (by searching for subgoals using linear lookahead). Once the goal vector is determined, the agent performs obstacle detection around its current position and updates the firing map using the Boundary Cell model. The Head Direction Calibration model receives the angular velocity and position of the agent to decode the head direction. If the cue is within the agent's range of view, the head direction can be corrected.
The Grid Cell model receives the agent's speed and fires accordingly, while the firing rate of the Place Cell model is computed based on the Grid Cell modules. When the agent enters

a new place, a new Place Cell is created, and the required information for the cognitive map is saved in the Prefrontal Cortex model, which includes Recency Cell, Topology Cell, and Reward Cell, as depicted in Figure 5.2. The system proceeds to the next iteration of exploration or navigation until the agent reaches the goal or the cognitive map is complete.

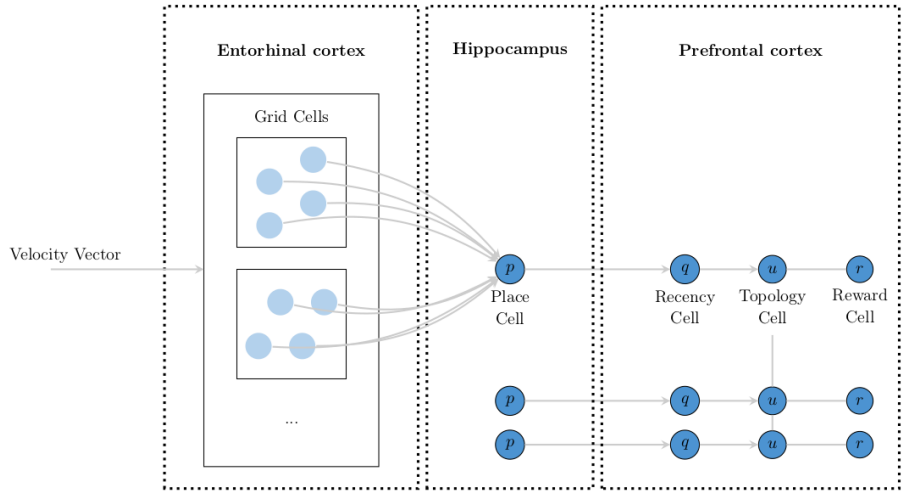

Figure 5.2: Detailed structure of prefrontal cortex cell. The computation of the firing rate of place cells is based on the firing of grid cells. Each place cell is connected to a prefrontal cortex cell model that includes three cells: a recency cell, which keeps track of visited places, a topology cell, which encodes the spatial relationships between locations, and a reward cell, which identifies the proximity of the goal [Eng21].

The firing rate of place cells is determined by the firing rate of grid cells. Each place cell is connected to a prefrontal cortex cell model, consisting of a recency cell, a topology cell, and a reward cell, which respectively store information about the agent's visited history, the neighborhood of locations, and the reward gradient. The next iteration of exploration or navigation begins, and this process continues until the agent reaches the goal or completes the cognitive map. The exploration phase involves the agent navigating through a predefined list of subgoals, as opposed to random exploration, which can be time-consuming and result in the exploration of irrelevant locations. The cognitive map is then used for navigation to reach the goal. In the following sections, we will provide detailed descriptions of the basic models used in our system.

5.2 Head Direction Calibration Model

This section focuses on the Head Direction Calibration model (HDC), which is composed of two main components: the basic HD model and the Calibration Circuit. See Figure 5.3 for an illustration of the HDC structure. The proposed HDC model is composed of two parts: the basic HD model, which is used for HD estimation in path integration and is reused from Amir[Sew20], and the Calibration Circuit (CC), which is used for correcting the error of HD estimation and is reused from Dominik[Nit21]. In this section, we will first introduce the individual components of the model, followed by a description of the method and algorithm for calculating the weights between rings of neurons. Once the connections between these rings are established, an initialization process is performed to obtain a stable activity peak in HD. This peak can be shifted upon receiving stimulus from shift layers, ECD rings, and ACD rings. The relevant directions can be decoded from the activity of these rings.

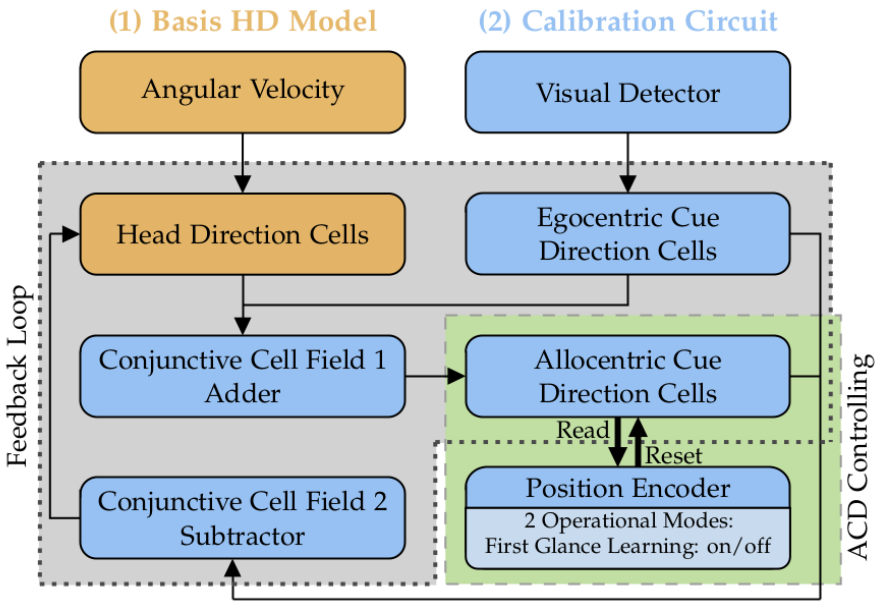

Figure 5.3: Structure of HDC model.(1) TThe fundamental component of the HDC model is the basic HD model, which estimates the head direction based on path integration and takes angular velocity as input. (2) The Calibration Circuit (CC) is responsible for correcting any errors in the HD estimation by incorporating cues. When a cue is detected by the Visual Detector, the Angle-Cue Distance (ACD) information is inferred by adding the HD and ECD information. The Position Encoder is then used to reset and correct the ACD information. Finally, the HD information is generated by subtracting the ECD information from the ACD information using a Subtractor component [Nit21].

5.2.1 Model Components

In this section, we will start by introducing the basic neuron model that is used in both the basic HD model and the components of the Calibration Circuit. Then, we will describe the individual components of the model shown in Figure 5.3.

Neuron Model

The spiking neurons are the fundamental building blocks for both the HD model of Amir and the CC of Dominik. They are used to simulate the behavior of biological neural networks. The activity of each neuron is determined by the summation of the weighted firing rates from other neurons in the network. The neuron model, adapted from Pereira [PB18] and Amir [Sew20], is described by the following Equation 5.1:

$$\tau \frac{df_i}{dt} = -f_i + \phi(I_i + \sum_{j=1}^{n} w_{ij} f_i) \tag{5.1}$$

with:
τ = time constant of firing rate dynamics
f_i = firing rate of neuron i (Hz)
f_j = firing rate of neuron j (Hz)
ϕ = single neuron transfer function
I_j = external input current to neuron i
w_{ij} = connection weight from neuron j to neuron i

The single neuron transfer function $\phi(x)$, reused from Pereira[PB18], Amir[Sew20] and Dominik[Nit21], is defined in Equation5.2 :

$$\phi(x) = \frac{r_m}{1 + exp(-\beta_t(x - h_0))} \tag{5.2}$$

with:
r_m = 76.2Hz
β_t = 0.82
h_0 = 2.46
τ = 20ms

The sigmoid function ϕ utilizes parameters obtained from recorded in-vivo neurons [PB18] and is reused in this work. The firing rate of the neuron approaches $\phi(x)$ over time, as stated in Equation 5.1. If $x = 0$, a neuron without external input has a firing rate of 8.95 Hz, as per Equation 5.2. The upper limit of $\phi(x)$ is the maximal firing rate r_m of 76.2 Hz.

Head Direction Attractor

The Head Direction (HD) model is composed of a HD attractor and two shift layers, as shown in the right part of Figure 3.2 (only one shift layer is shown). The HD attractor is a ring of n neurons, where each neuron is connected to other neurons and has a specific activity corresponding to the current head direction of the agent. The estimated HD can be inferred by the peak firing rate in the HD attractor. After initialization, a stable activity peak can be obtained in the HD attractor and can be shifted by the shift layers when they receive angular velocity. The neurons in the HD attractor can be numbered with $i \in \{0, ..., n-1\}$. Each neuron corresponds to a certain orientation, which is called the Preferred Firing Direction (PFD). The neurons' PFDs are equally spaced around the circle of 360° in counterclockwise direction and are given in the interval $[0, 2\pi)$. Thus, the neuron's PFD can be calculated by using Equation 5.3.

$$\theta_i = \frac{2\pi i}{n} \tag{5.3}$$

The firing rate of neuron i in the HD attractor depends on the difference between its PFD θ_i and the current HD θ_A. The angular difference can be calculated using Equation 5.4 and the firing rate of neuron i can be determined using Equation 5.5 [Nit21].

$$\Delta\theta_{iA} = \theta_A - \theta_i \tag{5.4}$$

$$f(\Delta\theta_{iA}) = A + B\mathrm{e}^{K\cos(\Delta\theta_{iA})} = 1.72 + 0.344\mathrm{e}^{5.29\cos(\Delta\theta_{iA})} \tag{5.5}$$

with:

i = index of neuron
n = number of neurons
θ_i= the preferred firing direction of neuron i
θ_A= actual head direction
f = firing rate of at angle difference $\Delta\theta_{iA}$ (Hz)
A = firing rate of neuron (Hz)
$B\mathrm{e}^K$ =peak firing rate (Hz)
K = sharpness parameter of the peak curve

The connections between neurons in the HD attractor are reciprocal and based on the difference between their PFDs. This creates a CANN where neighboring neurons have excitatory connections and distal neurons have inhibitory connections. The weights between neurons ensure that a stable activity peak is established after removing the stimulus during initialization. This stable activity peak can be shifted by the shift layers when the system receives angular velocity.

Shift Layers

To shift the activity peak in the HD attractor, two Shift Layers (SLs) are added to the network: Shift Layer Left (SLL) and Shift Layer Right (SLR). The SLL is used to shift the activity peak counterclockwise, and the SLR clockwise. They are parallel to the HD attractor, and each neuron in the HDC has a corresponding neuron in SLL and SLR. The activity profile of the SLs is similar to the HD's activity profile, but with a lower amplitude to tolerate the higher firing rate due to external stimulus. The weights from neurons in the HDC to the SLs are only half of the weight between neurons in the HD attractor. When external stimulus is applied to the SLs, the activity peak in the HD attractor will be shifted. Each neuron in the SLs has an excitatory connection in the shift direction and an inhibitory connection in the other direction. To ensure that the HDC is not impacted without external stimulus, the connections from the SLs to the HDC should cancel out if they have the same activity. The weights from neuron j of the SLs to neuron i of the HDC are determined by Equation 5.6 [Sew20].

$$W_{left}(i,j) = W(i,(j-o) \pmod n) - W(i,(j+o) \pmod n) \tag{5.6}$$

with:

n=number of neurons of HDCs
o=offset in neurons
j=index of neuron in shift layers
i=index of neuron in HDCs

The stimulus in the shift layers is induced by the angular velocity of the agent and can be computed using Equation 5.7 [Sew20].

$$s = 0.178124v \tag{5.7}$$

with:

v = absolute angular velocity (rad/s)
s = stimulus

When the agent turns left, only the SLL will receive a non-zero stimulus, while the SLR will have a zero stimulus. On the other hand, when the angular velocity is clockwise, only the SLR will receive a non-zero stimulus, while the SLL will have a zero stimulus. This allows for the activity peak in the HD attractor to be shifted in the appropriate direction.

Visual Detector

The Visual Detector is not a model consisting of neurons, but rather an abstract block used to detect visual cues. This block is defined by the field of view and detection range of the agent. When the agent moves, if a visual cue is within the field of view and detection range, the Visual Detector is activated and drives the ECDCs.

Egocentric Cue Direction Cells

The Egocentric Cue Direction Cells (ECDCs) are responsible for perceiving the cue information in the egocentric frame. The activity of these neurons depends on the agent's position and orientation. Like the HD attractor and shift layers, ECDCs also consist of n neurons equally spaced around a circle of 360°, which can be numbered with $i \in \{0, ..., n-1\}$. The angular position of neuron i within the interval $[0, 2\pi)$ can be calculated using Equation 5.8, while the firing rate of the neurons can be determined using Equation 5.10 [Nit21].

$$\alpha_i = \frac{2\pi i}{n} \tag{5.8}$$

$$\Delta\alpha_{il} = \alpha_l - \alpha_i \tag{5.9}$$

$$f(\Delta\alpha_{il}) = A + B\mathrm{e}^{K\cos(\Delta\alpha_{il})} = 1.72 + 0.344\mathrm{e}^{5.29\cos(\Delta\alpha_{il})} \tag{5.10}$$

with:

i = index of neurons
n = number of neurons
θ_i = angular position of ECDCs i
θ_l = a landmark's angular position in relation to the orientation of agent in egocentric frame
f = firing rate of at angle difference $\Delta\theta_{iA}$ (Hz)
A = firing rate of neuron (Hz)
$B\mathrm{e}^K$ =peak firing rate (Hz)
K = sharpness parameter of the peak curve

The Equation 5.10 describes the firing rate profile of the ECDCs. Neurons closer to the angular position of the landmark have a higher firing rate. An example is shown in the left part of Figure 5.4 [Nit21]. The rings consist of $n = 8$ neurons. The ECDCs represent the landmark in the egocentric frame, and the agent's orientation is always 0°. Neuron $i = 0$ represents the head direction of the agent, and neuron $i = 2$ represents the agent's left side (90°). The angular position of the landmark is approximately 292.5°. Therefore, neurons $i = 6, 7$ (in dark orange) are closer to the angular position of the landmark, and their firing rates are stronger than other neurons. The firing rate can be determined by the Equation 5.10. Using more neurons in the rings leads to more accurate firing rates.

Allocentric Cue Direction Cells

The Allocentric Cue Direction Cells (ACDCs) also represent the bearing of a visual landmark, but in an allocentric frame of reference. Like ECDCs, the numbering system for neurons is the same, except that β is used instead of α. The firing rate of ACDCs is determined by Equations 5.11, 5.12, and 5.13 [Nit21].

$$\beta_i = \frac{2\pi i}{n} \tag{5.11}$$

$$\Delta\beta_{il} = \beta_l - \beta_i \tag{5.12}$$

$$f(\Delta\beta_{il}) = A + B\mathrm{e}^{K\cos(\Delta\beta_{il})} = 1.72 + 0.344\mathrm{e}^{5.29\cos(\Delta\beta_{il})} \tag{5.13}$$

with:

i = index of neuron
n = number of neurons

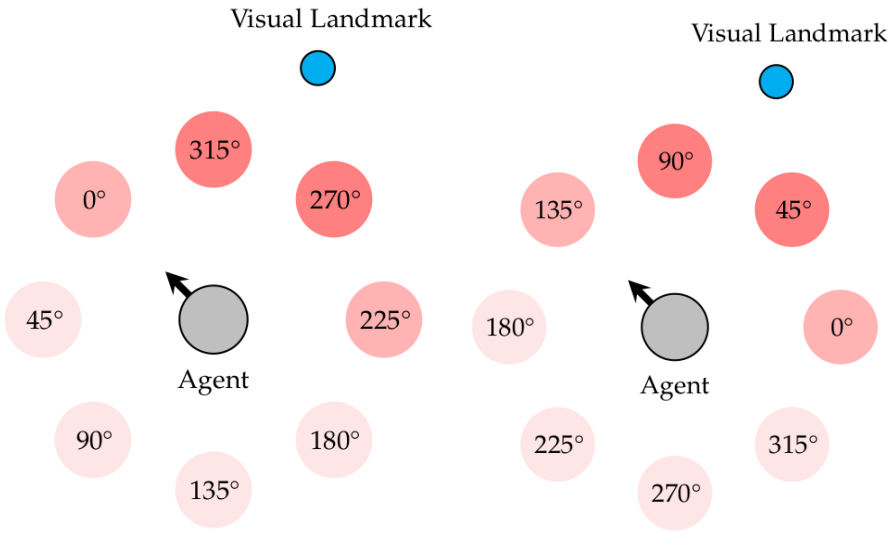

Figure 5.4: Comparison of ECDCs (left) and ACDCs (right) with $n = 8$. (Left)The ECDCs represent the landmark location in the egocentric frame of reference, with the agent's orientation fixed at 0°, while the angular position of the landmark depends on the agent's position and orientation. (Right)The ACDCs represent the landmark location in the allocentric frame of reference, with the east of the agent serving as the reference point at 0°, and the angular position of the landmark depending solely on the agent's position [Nit21].

$\beta_i =$ angular position of ACDCs i
$\beta_l =$ a landmark's angular position in relation to the orientation of agent in allocentric frame
f = firing rate of at angle difference $\Delta\beta_{iA}$ (Hz)
A = firing rate of neuron (Hz)
$B e^{K}$ =peak firing rate (Hz)
K = sharpness parameter of the peak curve

The firing rate of ACDCs is determined by Equation 5.13, which describes the target firing profile. For instance, in a circle of 360° containing $n = 8$ ACDCs, the neurons are equally spaced and independent of the agent's orientation (see right part of Figure 5.4). The neuron $i = 0$ always corresponds to the east of the agent (0°). For an agent orientation of 135° and a landmark angular position of approximately 67.5° in allocentric frame, neurons $i = 1$ and $i = 2$ fire stronger than the other neurons due to their smaller angular distance to the landmark ($\Delta\beta_{il}$), while the other neurons have larger angular distances and smaller firing rates, in accordance with Equation 5.13.

Conjunctive Cell Field 1: Adder

Figure 5.5 illustrates how the angular position of a visual landmark can be represented either in allocentric or egocentric frame.
The ACD of a landmark can be determined based on the agent's HD and ECD using the Equation 5.14.

$$\beta_l = \theta_A + \alpha_l \pmod{360°} \tag{5.14}$$

where θ_A is the HD, α_l is the angular position of landmark in egocentric frame and β_l is the angular position of landmark in allocentric frame. In order to ensure that the ACD is within the interval of $[0°, 360°)$, the modulo operation is applied to the equation that relates the ACD with the HD and ECD of the landmark. To simulate this function, the Conjunctive Cell Field 1:Adder (CCF 1) from the work of Dominik is used. CCF 1 is a square sheet of neurons, numbered from $i = 0$ to $i = n - 1$ in a left-to-right, top-to-bottom manner, as shown in the

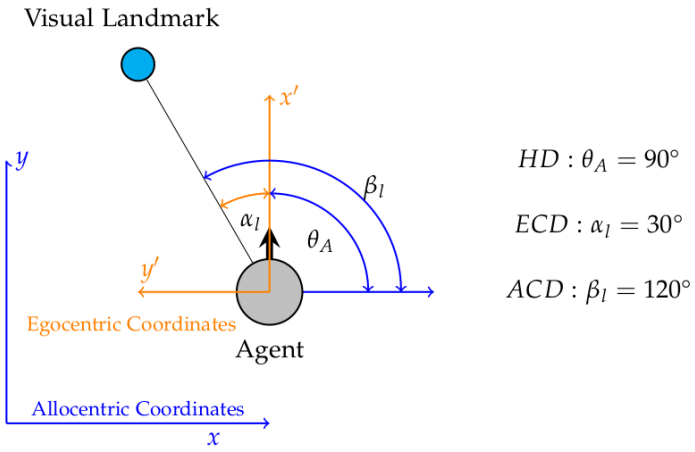

Figure 5.5: Angular position of visual landmark in different frames. The HD of an agent can be inferred based on the information from both ACD and ECD. In addition, ACD information can be calculated based on the other two types of information [Nit21].

left part of Figure 5.6. Each row and column consists of $\sqrt{n}$ neurons and is associated with two inputs: the HD θ_A of the agent and the ECD α_l of the visual landmark. The activity of the neuron i is determined by Equations 5.15, 5.16, and 5.17, as described in the works of Amir and Dominik [Sew20][Nit21].

$$\alpha_i = \frac{2\pi}{\sqrt{n}}(i \mod \sqrt{n}) \tag{5.15}$$

$$\theta_i = \frac{2\pi}{\sqrt{n}}(\sqrt{n} - 1 - (i \operatorname{div} \sqrt{n})) \tag{5.16}$$

$$f(\Delta\alpha_{il}, \Delta\theta_{iA}) = A + B\mathrm{e}^{0.5K(\cos(\Delta\alpha_{il})+\cos(\Delta\theta_{iA}))} = 0.0504\mathrm{e}^{2.645(\cos(\Delta\alpha_{il})+\cos(\Delta\theta_{iA}))} \tag{5.17}$$

with:

α_i = egocentric cue direction of neuron i
θ_i = head direction of neuron i
α_l =a landmark's angular position in relation to the orientation of agent in egocentric frame
θ_A = head direction of the agent in allocentric frame
f = firing rate at HD angle difference $\Delta\theta_{il}$ and ECD angle difference $\Delta\alpha_{il}$ (Hz)
A = lowest firing rate (Hz)
$B\mathrm{e}^K$ =peak firing rate (Hz)
K = sharpness parameter of the peak curve

The relationship expressed in Equation 5.14 can be implemented through neurons using the CCF 1. To illustrate this, consider the example shown in the left part of Figure 5.6. The CCF 1 comprises $n = 36$ neurons arranged in 6 rows and columns. Given an ECD of $\alpha_l = 30°$ and an HD of $\theta_A = 90°$, the resulting activity peak moves along the diagonal corresponding to $\beta_l = 120°$ with neuron $i = 2$. If the agent maintains its position but changes its orientation, such as by turning 30° to the right, the HD decreases to $\theta_A = 60°$ while the ACD increases to $\alpha_l = 60°$. The activity peak still moves along the diagonal, and the ACD remains the same. If the agent moves upwards without turning, the HD remains unchanged while both the ECD and ACD increase simultaneously. This is why CCF 1 is also known as an "Adder".

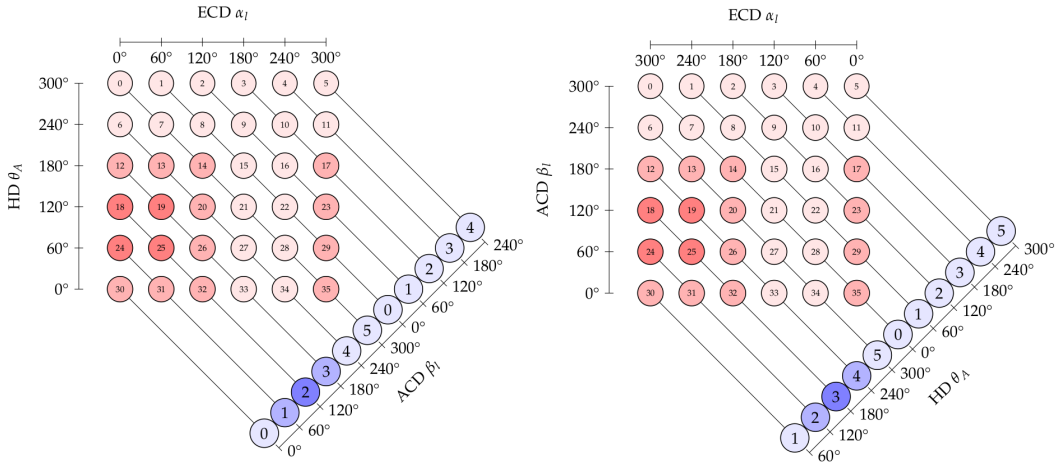

Figure 5.6: CCF1: Adder(left) and CCF2: Substractor(right) with $n = 36$ [Nit21].

Conjunctive Cell Field 2: Substactor

According to Figure 5.14, it is also possible to estimate the HD based on the ACD and ECD. Specifically, Equation 5.18 can be used to calculate the HD θ_A from the ACD β_l and ECD α_l.

$$\theta_A = (\beta_l - \alpha_l + 360°) \pmod{360°} \tag{5.18}$$

CCF 2 is used to reset the HD signal based on the ACD and ECD as inputs. It has the same sheet of neurons and numbering system as CCF 1 but with different inputs. The firing pattern of neuron i is determined by Equations 5.19, 5.20, and 5.21 [Nit21].

$$\alpha_i = \frac{2\pi}{\sqrt{n}}(\sqrt{n} - 1 - (i \operatorname{div} \sqrt{n})) \tag{5.19}$$

$$\beta_i = \frac{2\pi}{\sqrt{n}}(\sqrt{n} - 1 - (i \operatorname{div} \sqrt{n})) \tag{5.20}$$

$$f(\Delta\alpha_{il}, \Delta\beta_{il}) = A + Be^{0.5K(\cos(\Delta\alpha_{il})+\cos(\Delta\beta_{il}))} = 0.0504e^{2.645(\cos(\Delta\alpha_{il})+\cos(\Delta\beta_{il}))} \tag{5.21}$$

with:

α_i = egocentric cue direction of neuron i
β_i = allocentric cue direction of neuron i
α_l =a landmark's angular position in relation to the orientation of agent in egocentric frame
β_l = a landmark's angular position in relation to the agent in allocentric frame
f = firing rate at ACD angle difference $\Delta\beta_{il}$ and ECD angle difference $\Delta\alpha_{il}$(Hz)
A = lowest firing rate (Hz)
Be^K =peak firing rate (Hz)
K = sharpness parameter of the peak curve

To explain CCF 2, we can refer to the example shown in the right part of Figure 5.6. In this example, the sheet of neurons consists of 36 neurons, which are arranged in 6 rows and columns to represent a circle of 360° with a resolution of 60°. When ECD $\alpha_l = 270°$ and ACD $\beta_l = 90°$, neurons $i = 18, 19, 24, 25$ have higher activity than other neurons, and their activities are summed up via the diagonal, leading to the highest firing rate in neuron $i = 3$ in HD. Therefore, CCF 2 enables the decoding of HD based on ECD and ACD inputs.

Position Encoder

The Position Encoder (PE) is an artificial model, and here we will adapt and use Dominik's model for this purpose [Nit21]. The PE model aims to reset ACD using cue information, and it has two modes: First Glance Learning (FGL) on and off. The FGL "on" mode has been shown to correct ACD errors better than FGL "off," so we will focus only on the FGL "on" mode.
Firstly, a matrix containing position information is created with the visual landmark as the center, as shown in Figure 5.7. The matrix's range is determined by its size, and its resolution is dependent on the number of matrix elements. For example, if the matrix size is 4 and the number of matrix elements is 16, then the maximum detection range is 2, and the resolution is 1 x 1 units.
Secondly, when the agent enters the matrix's range and the landmark is within the agent's field of view, the agent stores information about the position, the ACD, and the distance to the landmark. When the agent enters the range again and the landmark is within its field of view, the ACD can be estimated based on the previously stored information. The estimated ACD is calculated by adding the vector between the current position and the stored position of the agent with the previous stored landmark vector. This estimated ACD can then be used to reset the ACD information, which can also reset the ACDC's activity via CCF.

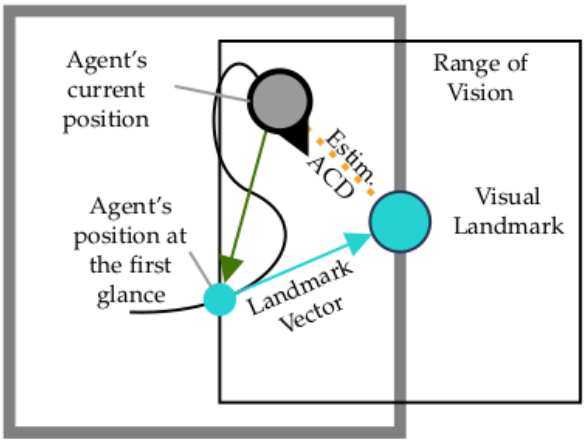

Figure 5.7: Illustration of the Position Encoder(PE). The PE stores information about the agent's position and the landmark's vector when the agent first enters the range and the visual landmark is in the agent's field of view. This stored information will be used later to decode the ACD.

5.2.2 Connection Weights

In this section, we will describe the method and algorithm used to calculate connection weights for many of the SNN models in the HDC model.

Connections Overview

The connectivity patterns of the models comprised of neurons are illustrated in Figure 5.8. Among these models, SLL, SLR, and HD form the fundamental components of the HD model, while ECD, CCF 1, ACD, and CCF 2 are parts of the Calibration Circuit (CC). Since the CCFs are 2D sheets of neurons that have connections with ECD, ACD, or HD, the algorithm used

to calculate the CC models differs from that used for basic HD models. In the following subsections, we will first explain the approach and algorithm used for basic HD models.

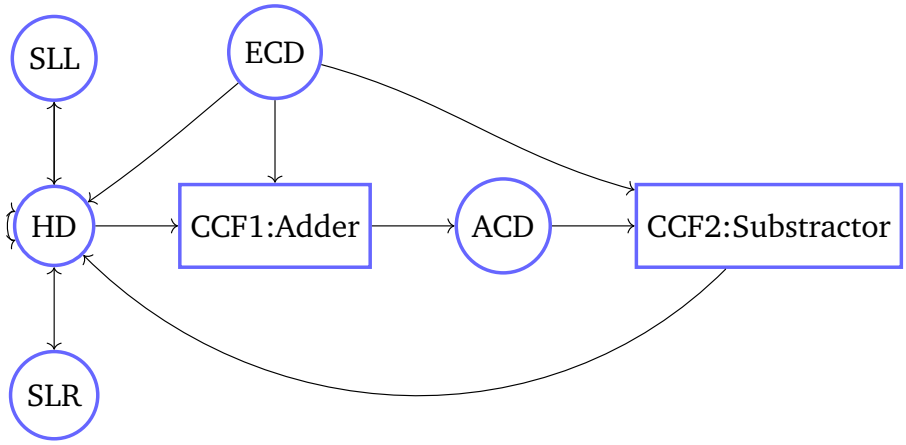

Figure 5.8: The conncenction between rings of neurons. The circles represent 1D ring neurons, specifically the HD neurons. The rectangles represent 2D sheets of neurons, such as CCF1. The lines with one arrow represent unidirectional connections, for example, the connection from ECD to HD. The lines with two arrows represent directional connections, such as the connection between SLL and HD.

Weights for Basic HD model

In the HDC model, each HD neuron is connected bidirectionally with all neurons in the SLL and SLR layers, and also with other neurons in the HD layer. However, there is no direct connection between the SLL and SLR layers. The connection weights between HDCs are similar to those between SLL or SLR and HDC, except that the weights between HDCs have smaller amplitude or offset.
To calculate the connection weights, we first compute the general weights using Equation 5.22 [Zha96]. We will then explain the detailed connection weights.

$$\hat{w}_n = \frac{\hat{u}_n \hat{f}_n}{\lambda + |\hat{f}_n|^2} \tag{5.22}$$

with:

$\hat{w}_n =$ n_{th} Fourier coefficient of the connection weights vector
$\hat{u}_n =$ n_{th} Fourier coefficient of the input current vector
$\hat{f}_n =$ n_{th} Fourier coefficient of the target firing vector
$\lambda =$ regularization parameter to control the flatness of the weight function

The target firing profile used in this work is based on the previous work of Amir[Sew20]. It depends on the preferred firing direction (PFD) of each neuron in the HD network, and it is given by Equation 5.23.

$$f(\theta_i) = A + Be^{K\cos(\theta_i)} = 1.72 + 0.344e^{5.29\cos(\theta_i)} \tag{5.23}$$

The values of K and A were determined in the work of Zhang [Zha96]. A is the background firing rate, and K determines the width of the activity peak. The maximal firing rate is typically in the range of 5 Hz to 120 Hz, and we chose 70 Hz, which is biologically plausible [Sew20]. By solving the equation $A + Be^K = 70$, we can calculate that $B = 0.344$.

Algorithm 1: Calculate weights with given λ

input :
n:number of neurons
θ_i:preferred directions
λ: flatness parameter
A, B, K:parameters for target firing rates
output:
W:a list of weights indexed by distance in intervals between HDCs
// Calculate target firing rate F
for $i = 0...n-1$ **do**
  $F[i] := A + Be^{K\cos(\theta_i)}$;
end
// Calculate U from F
for $i = 0...n-1$ **do**
  $U[i] := \phi^{-1}(F[i])$;
end
// Fast Fourier Transfoms
$\hat{F} := fft(F)$;
$\hat{U} := fft(U)$;
// Calculate the fourier coefficients of$W$ using Equation 5.22
for $i = 0...n-1$ **do**
  $\hat{W}[i] = \frac{\hat{U}[i]\hat{F}[i]}{\lambda + |\hat{F}[i]|^2}$;
end
// apply the inverse FFT to get W
$W := ifft(\hat{W})$;
return W;

Using the given value of λ and the target firing rate f, we can calculate the general weights using Algorithm 1. Once the general weights have been calculated using Algorithm 1, the detailed connection weights can be obtained as shown in Table 5.1. The connection weights from HDC to SLL or SLR are only half of the weights between neurons in the HD network to leave room for higher firing activity caused by stimuli. An offset of $o = 5$ is used for the weights from SLL or SLR to HDC to ensure that the two shift layers cancel out when they have the same activity and have no impact on the HDC network if internal stimulus arises.

Connection from	Connection to	Connection weights
neuron j in HDC	neuron i in HDC	W_{ij}
neuron i in HDC	neuron i in SLL or SLR	0.5 W_{ij}
neuron j in SLL	neuron i in HDC	$W_{left}(i,j)$ in Equation 5.6
neuron j in SLR	neuron i in HDC	$W_{right}(i,j) = -W_{left}(i,j)$

Table 5.1: Connection weights in basic HD model

Weights for Calibration Circuit

To calculate the general weights for the Calibration Circuit (CC) components, we will use an algorithm from the work of Dominik[Nit21]. The Algorithm 1 is only applicable to connections between neurons of 1D rings, whereas CCFs are 2D sheets of neurons. The firing

function in neurons of CCFs depends on two inputs, as shown in Equations 5.17 and 5.21. Therefore, the 2D activity profile in neurons of CCFs can be divided into two 1D activity profiles. If only one input projects onto a CCF, a 2D activity band profile, instead of a peak profile, will be obtained. For example, in the right part of Figure 5.6, if only ACDCs with peak profile at 60° project onto CCF2, all neurons $i = 24$ to 29 will be more active than other neurons. If the second input ECDC with peak profile at 240° projects onto the CCF2, then the desired 2D activity profile with a peak at neuron $i = 25$ will emerge, which means both activity band's maximum firing rates meet at neuron $i = 25$. Therefore, we will use the algorithm from the work by Dominik [Nit21].
Firstly, a measure value containing the regularization parameter λ is designed to measure the error between the desired input and actual input, which is shown in Equation 5.24.

$$E = \sum_{n=1} |\hat{u}_n - \hat{w}_n \hat{f}_n|^2 + \lambda \sum_{n=1} |\hat{w}_n|^2 \tag{5.24}$$

with:

$E =$ error measurement value
$\hat{w}_n =$ n_{th} Fourier coefficient of the connection weights vector
$\hat{u}_n =$ n_{th} Fourier coefficient of the input current vector
$\hat{f}_n =$ n_{th} Fourier coefficient of the target firing vector
$\lambda =$ regularization parameter to control the flatness of the weight function

The Algorithm 2 [Nit21] is used to determine the weight distributions for Calibration Circuit (CC) components based on an error measurement value that includes a regularization parameter λ. This algorithm is a general method that takes into account the desired firing rates F of pre-synaptic neurons and the desired input current U of post-synaptic neurons, which vary depending on the type of connection between neurons. The desired firing rates are calculated using two types of firing rate equations, namely Equation 5.23 and Equation 5.25 from the work by Dominik[Nit21]. The desired input currents and firing rates for each type of neuron are summarized in Table 5.2.
Different scaling factors are used for the transformation between firing rate and input current, which is represented by $u = \phi^{-1}(f)$. For instance, the ECDC and ACDC neurons have the same firing rate equation as in Equation 5.23 when projecting firing onto CCF1, and the scaling factor for input current in CCF1 is chosen to be 0.5. A factor of 0.2 is used for the ACD neurons to achieve a small activity profile in the exploration phase, and a factor of 0.1 is used to control the activity peak position without introducing too much bias into HDC's activity profile.
As there are reciprocal connections between neurons in the HD attractor, the HDC signal cannot be inferred using Equation 5.23 without a calibrated signal from CCF2. Therefore, a measured HD signal in the simulation is used for the connection from HD to CCF1.

$$f(\theta_i) = A + B e^{K \cos(\theta_i)} = 0.0504 e^{5.29 \cos(\theta_i)} \tag{5.25}$$

5.2.3 Initialization

To ensure a stable activity peak centered at 0° in the HD network, the network is initialized with a specific stimulus for each neuron i, defined as $I_i = \phi^{-1}(0.5 f(\theta_i))$, where f is the target firing profile determined by Equation 5.23. Initially, the stimulus is applied to the HD network for a specific duration t_{stim}, and the firing rates are updated using the explicit Euler

Pre-synaptic neurons	Pos-synaptic neurons	Firing profile F	Equation of f in F	Input Current U	Equation of f in U
ECDCs or ACDCs	CCF 1	Equation 5.23	Equation 5.23	$0.5\phi^{-1}(f)$	Equation 5.25
CCF 1	ACDCs	depend on $0.5\phi^{-1}(f)$	Equation 5.25	$0.2\phi^{-1}(f)$	Equation 5.23
CCF 2	HDCs	depend on $0.5\phi^{-1}(f)$	Equation 5.25	$0.1\phi^{-1}(f)$	Equation 5.23
HDCs	CCF 1	Measurement from simulation			

Table 5.2: Input current U and target firing profile F for calculating weights in Calibration Circuit

Algorithm 2: Calculate weights for Calibration Circuit

input :
- λ: flatness parameter
- n: number of neurons
- Λ: vector containing several flatness parameters λ
- F: desired firing profiles of n pre-synaptic neurons
- U: desired input currents of n post-synaptic neurons
- E: variable storing the smallest summation of squared error for λ

output:
- W_{opt}: list containing the optimal connection weights

```
// initialize E with infinity
```
$E := \infty$;
```
// Fast Fourier Transfoms
```
$\hat{F} := fft(F)$;
$\hat{U} := fft(U)$;
for λ in Λ **do**
  for $i = 0...n-1$ **do**
    ```
    // calculate the fourier coefficients of W
    ```
    $\hat{W}[i] = \frac{\hat{U}[i]\hat{F}[i]}{\lambda + |\hat{F}[i]|^2}$;
  end
  ```
  // compute the inverse fourier transformation
  ```
  $W := fft(\hat{W})$;
  ```
  // compute the sum of squared errors and store it
  ```
  $X := \sum_{i=0}^{n-1}(U[i] - W[i]F[i])^2 + \frac{1}{\lambda}$;
  if $E > X$ **then**
    $W_{opt} := W$;
  end
end
```
// return the weight list that face the smallest sum of errors
```
return W_{opt};

formula. Subsequently, the stimulus is removed for a given time period t_{settle}. The changes in firing rates during a time interval $t_{interval}$ are summed up and stored in δ. This process is repeated until the changes in firing rates are below a predetermined threshold ϵ.

5.2.4 Apply Stimulus and Decode Directions

After the initialization of the HD network, the HDC model can receive input current from various sources, including SLL or SLR, ECD ring, and ACD ring caused by external stimuli. The potential input currents are shown in Table 5.3. When the HDC model receives these input currents, the firing rate is updated using explicit Euler formula in a step-wise simulation. The firing rates of the neurons in HDC, ACDC, and ECDC can be read out and used for decoding of the HD, ACD, and ECD with the Equations 5.26, 5.27, and 5.28.

$$\theta_i = \frac{2\pi}{n} i \tag{5.26}$$

$$v_i = \begin{pmatrix} f_i \cos\theta_i \\ f_i \sin\theta_i \end{pmatrix} \tag{5.27}$$

$$\gamma = \arctan(2\sum v_i) \tag{5.28}$$

with:

n= the number of neurons in the corresponding ring
$i = 0, ..., n-1$
f_i= firing rates of the neurons in corresponding ring after step simulation
γ= decoded direction

Situation	Input current	Relevant Equation	Apply to rings
angular velocity in shift direction of SLs	$U = s$	s from Equation 5.7	SLL or SLR
Cue in range and in field of view of agent	$U = \phi^{-1}(f)$	f from Equation 5.10	ECD ring
ECD ring active and ACD ring active	$U = 0.8\phi^{-1}(f)$	f from Equation 5.13	ACD ring

Table 5.3: Input currents for HDC model

5.3 Boundary Cell Model

In the Boundary Cell (BC) model used in this work, rate-coded neurons are employed, as in the original model proposed by Camillo [Hey21]. As the information about the head direction of the agent is already contained in the HDC model, the method described in Camillo's work to transfer the BC from egocentric to allocentric frame is not used here.
To build the firing map for the BC, polar coordinates are used. Each BC i has a preferred firing direction ϑ_i and a preferred firing distance p_i. When the boundaries are located at the position (p, ϑ), the firing rate r_i of the i_{th} BC can be calculated using Equations 5.29 and 5.30 [Hey21]. The model consists of n neurons used to build a firing map based on the detection of boundaries. The parameters σ_ϑ and σ_p are used to define the spatial dispersion of the firing, with the former being a constant that sets the angular tuning width, and the latter being linear with the preferred distance and used to set the sensitivity of the firing to distance.

$$\sigma_p(p_i) = (p_i + 8) * 0.08 \tag{5.29}$$

$$r_i = \frac{1}{p} * \exp(-(\frac{\vartheta_i - \vartheta}{\sigma_\vartheta})^2) * \exp(-(\frac{p_i - p}{\sigma_p})^2) \tag{5.30}$$

with:
(p, ϑ)= distance and angular of the boundary in polar coordinate
(p_i, ϑ_i)= preferred distance and angular of i_{th} BC in polar coordinate
σ_ϑ=0.2236,parameter for controlling angular tuning width
σ_p=parameter for controlling tuning distance

5.4 Grid Cell Model

The GC model used in this work is adapted from the work by Tim [Eng21]. The model consists of a single GC module, which is described first, followed by the combination of modules to form the complete GC model.

5.4.1 Single Grid Cell Module

The GC model from the work by Tim [Eng21] consists of a 2D sheet of neurons, with each module consisting of n x n neurons, as shown in Figure 5.9. The sheet is connected at the North (N) and South (S) edges, as well as the East (E) and West (W) edges, enabling the calculation of the distance between two neurons. Each neuron is assigned a preferred firing direction, represented by a unit vector $\hat{e}_{\theta_i}$ corresponding to the direction $\theta_i \in [W, N, E, S]$ chosen based on the index $(2(y \mod 2) + x \mod 2)$.
The coordinates of a neuron i in the sheet can be represented as x_i in sheet coordinates, and the connection weight from neuron i to neuron j is determined by the recurrent connectivity profile given in Equation 5.31. To update the firing rate of a grid cell, we use the weight vector $w^{rec}_{x_i - \hat{e}_{\theta_i}}$ to calculate the change in firing rate as shown in Equation 5.32.

$$rec(d) = e^{-\gamma d} - e^{-\beta d} \tag{5.31}$$

with:
λ=15
$\beta = \frac{3}{\lambda^2}$
γ= 1.05β
d= the shortest distance between two neurons in sheet

$$\tau \frac{ds_i}{dt} + s_i = f(sw^{rec}_{x_i - \hat{e}_{\theta_i}} + B_i) \tag{5.32}$$

with $dt = 10ms, \tau = 100ms, f(x) = max(0, x)$, s is the previous external input B_i, which is expressed by the Equation 5.33.

$$B_i = 1 + g_m \alpha \hat{e}_{\theta_i} v \tag{5.33}$$

With $\alpha = 0.10315$, the input velocity vector is denoted as v, s represents the firing value at the end of the previous time step, and g_m is the scaling factor for the module, which will be described in the following section.

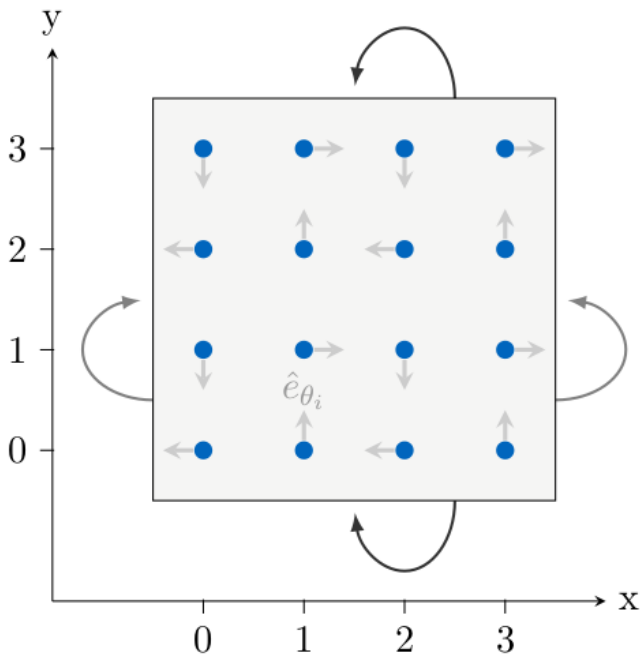

Figure 5.9: Illstration of n x n neuron sheet of Grid Cell Module with $n = 4$. All neurons in the GC module are connected along both the east-west and north-south directions. Each neuron is associated with a preferred firing direction $\hat{e}_{\theta_i}$, which can be in any of the four cardinal directions: north, west, east, or south.

5.4.2 Combining Grid Cell Module

We utilize a GC model that combines six GC modules ($M = 6$), each with a different scaling factor g_m, which is derived from the work of Tim [Eng21] and falls within the range of $g_m \in [0.2, 2.4]$ with $m \in [1, M]$. The minimum factor $g_{min} = 0.2$ is used to ensure an unambiguous firing pattern of the GC in the environment, while the maximum factor g_{max} determines the precision. The scaling factors g_m can be calculated using Equations 5.34 and 5.35.

$$R = \sqrt[M-1]{g_{max}/g_{min}} \tag{5.34}$$

$$g_m = g_{min} R^{m-1} \tag{5.35}$$

After constructing the GC model, we apply a random velocity input to the model for 1000 time steps with a time step size of $dt = 10ms$ to generate the hexagonal pattern. This pattern can be shifted by applying the velocity vector of the agent in the simulation. The scaling factor g_m determines the amplitude of the shift, and a properly chosen factor can ensure an unambiguous firing pattern of the GC model. The firing rate of the GC serves as the input for calculating the firing rate of Place Cells, which will be described in the next section.

5.5 Place Cell Model

Place Cells are responsible for firing at unique locations in the environment. This characteristic is utilized to store place information as the agent moves through the environment. In this work, the PC model developed by Tim [Eng21] is utilized. This model is a unidirectional connection model, where a new PC is created and assigned a spiking value determined by the GC firing when the agent enters a new location. All neurons with a spiking value greater

than 0.1 have a connection to the newly created PC with a connection weight of 1, as shown in Figure 5.2. The spiking value of the PCs is determined by the normalized GC spiking value $s/|s|$ and the GC connectivity profile c_i. With the given number of modules $M = 6$, the spiking value of the i_{th} PC can be expressed by Equation 5.36 [Eng21].

$$p_i = \frac{1}{M} \sum_{m=1}^{M} \frac{s_m c_{m|i}}{s_m} \tag{5.36}$$

When the agent enters a place field, the spiking value of the corresponding PC is set to 1. As the agent moves away from the center of the place field, the spiking value of the PC decreases. When the highest spiking value p_i among all PCs is below a threshold of 0.85 or the agent reaches its goal, a new PC is created for the new location with the spiking value determined by the normalized GC spiking $s/|s|$ and the GC connectivity profile c_i, as described in the previous Equation 5.36.

5.6 Cognitive Map

The prefrontal cortex cell model used in this work is based on the concept proposed by Erdem and Hasselmo [EH12] and was originally designed by Tim [Eng21]. The model consists of three types of cells: recency cells, topology cells, and reward cells. Each PC model is associated with a set of these three cells.

5.6.1 Recency Cells

At each time step, the current place cell with a spiking value p_i higher than a threshold of $p_{threshold} = 0.58$ is identified as the active PC. The recency cell linked to the active PC is assigned a value of $q_i = 1$, while the values of the other recency cells linked to inactive PCs decay over time according to Equation 5.37 [Eng21]. The recency cell thus stores the history of the agent's visited places.

$$q_i = \epsilon^{-\lambda t} \tag{5.37}$$

with $\epsilon = 2$ and $\lambda = 0.01/dt$. Unlike the topology cell and reward cell, which are updated only when the agent enters a new or different place field, the recency cell is updated at every time step, regardless of whether the agent enters a new place field or not.

5.6.2 Topology Cells

The topology cell stores information about the adjacent place fields. A bidirectional connection is created between topology cells when the corresponding place fields are neighbors. This topology is updated whenever the agent enters a new or different place field. A new matrix is created using Equation 5.38 [Eng21].

$$T_{new} = H(q^T - \delta)H(q - 1) \tag{5.38}$$

where $H()$ is the Heaviside step function with $H(0) = 1$ and $\delta = 0.5$. The first part of the equation, $H(q^T - \delta)$, is used for the previous visited cells, while the second part, $H(q - 1)$, is used for the current cell. The resulting new topology matrix, T_{new}, is created using Equation 5.38 and added to its transposed matrix and the previous matrix. The resulting matrix, $T = T \vee T_{new} \vee T_{new}^T$, represents the overall connections in the topology cells, where $T_{ij} = T_{ji}$ if there is a connection between the i_{th} and j_{th} topology cells.

5.6.3 Reward Cells

The reward cell is an important component for storing the location of rewards during exploration and helping the agent navigate towards them. The model used for the reward cell is borrowed from the work of Tim [Eng21], which was in turn inspired by Erdem and Hasselmo [EH12]. The reward cell has two functions: first, it remembers the reward value by setting $r_i = 1$ for the reward cells linked to the place field where the reward is located. Second, it creates a reward gradient that decreases as the agent moves away from the actual goal. When the agent enters a new place field, the corresponding reward value is stored in the reward cell. When the agent visits a different or new place field, the reward cell is updated. To update the reward cell, the reward is set to one for the actual goal and zero for other reward cells. Based on the topology cells and previous reward cells, we can propagate the reward through the network to the n_{th} neighborhood using Equation 5.39. In this equation, the k_{th} neighborhood reward cell has a reward value of $r_i = 1/(k+1)$ [Eng21].

$$r_i^k = max(\frac{1}{k+1} H(r^{(k-1)}T)_i, r_i^{k-1}) \tag{5.39}$$

where $H()$ is the Heaviside function and $k \in [0, n]$. n is the maximum number of neighborhoods that reward information can diffuse through and is typically chosen based on the size of the environment or the number of place cells. A larger environment or a larger number of place cells would require a larger n to ensure that the reward information can effectively propagate through the network.

5.7 Exploration

After initializing the basic models, we can begin the simulation which consists of an exploration phase and a navigation phase. The exploration phase is responsible for creating place cells in the PC model and storing information for the recency, topology, and reward cells. This information is used to build the cognitive map for the navigation phase. At the start of each iteration of the simulation loop, a sub-goal is identified. Rather than selecting a random location as the sub-goal which can result in excessive search times, we use the "linear sunburst" approach described by Tim [Eng21]. This involves using a predefined list of locations which cover all possible locations to find the goal as sub-goals during the exploration phase. The basic HDC model, Boundary Cell model, and GC model work similarly in both exploration and navigation phases. However, our focus is on how the cognitive map can be utilized during the navigation phase, and thus our work primarily focuses on this aspect.

5.8 Navigation

After the cognitive map has been created during the exploration phase, the navigation phase begins with the initialization of the basic models. An overview of the navigation process is depicted in Figure 5.10. The first step in navigation is to determine a goal vector for the agent, which is computed by the navigation phase controller.

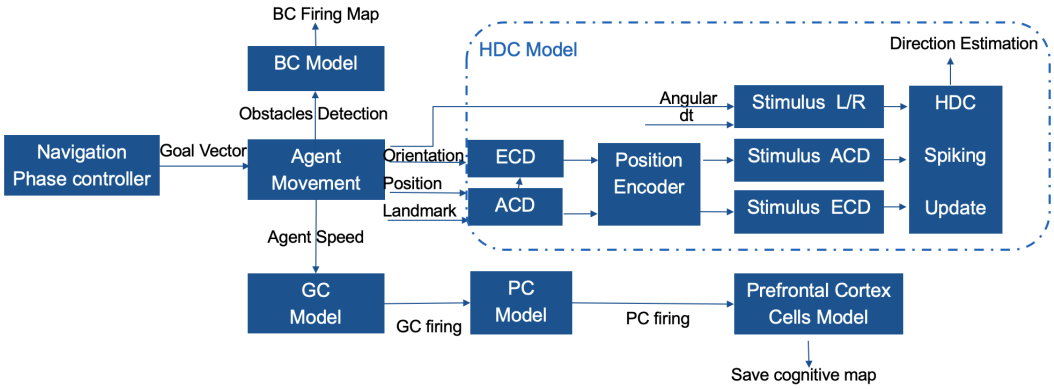

Figure 5.10: Illustration of detailed Navigation. The goal vector is computed by the navigation phase controller and then sent to the agent. In addition, the results of obstacle detection are used to create a firing map for the Boundary Cell model, and the agent's speed is fed to the Grid Cell model. The orientation, position, and location of landmarks are sent to the HDC model for decoding the Head-Direction (HD), Alocentric-Cue (AC), and Egocentric-Cue (EC) information, respectively.

5.8.1 Navigation Phase Controller

The navigation phase controller is responsible for calculating the next goal vector for the agent based on the current situation. This process is illustrated in Figure 5.11. At the beginning of each iteration, the goal vector is determined based on three situations: topology-based navigation where the previous sub-goal has been reached ($d < 0.3$), vector-based navigation where 80% of the original goal vector length has been covered, and all other situations.
For the third situation, the goal vector is updated simply by subtracting the distance already traveled. For the first and second situations, a linear lookahead method is used to calculate the new goal vector. This involves simulating a virtual movement of the agent at a given velocity, causing GC spiking, and finding the reward location based on the PC spike value. In the second situation, the lookahead algorithm is applied along two axes to find a new goal vector for vector-based navigation. In contrast, the directed lookahead algorithm is used to compute a sub-goal for topology-based navigation based on the detection angles.
At the start of the simulation, the agent is initialized with a vector-based navigation, and the navigation strategy is updated based on the detection of obstacles using 16 sensors around the agent. The agent's decision for a topology-based or vector-based navigation depends on the result of the obstacle detection.

Obstacles Detection

In this study, the environment is equipped with $n = 16$ sensors, which divide the 360° circle into 16 angles for obstacle detection. The detection range is set to 2m in our maze environment, and the agent takes a backup step if the distance to the nearest obstacle is less than 0.3m. To prevent collisions, we check the obstacles in the heading direction and goal vector direction, and if the distance to the obstacles is less than 0.6m, the agent switches to topology-based navigation. The obstacles detection process is illustrated in Figure 5.12 [Eng21]. A direction is considered blocked if the distance to obstacles in that direction is less than 1.3m, or if the distance to obstacles in neighboring directions is less than 0.9m. Taking neighboring directions into consideration is important to prevent collisions when the agent is navigating through corners. If all directions are unblocked, the agent switches to vector-based navigation to find a new goal vector. In topology-based navigation, the directed lookahead algorithm is used to find the sub-goal, which will be explained in the following

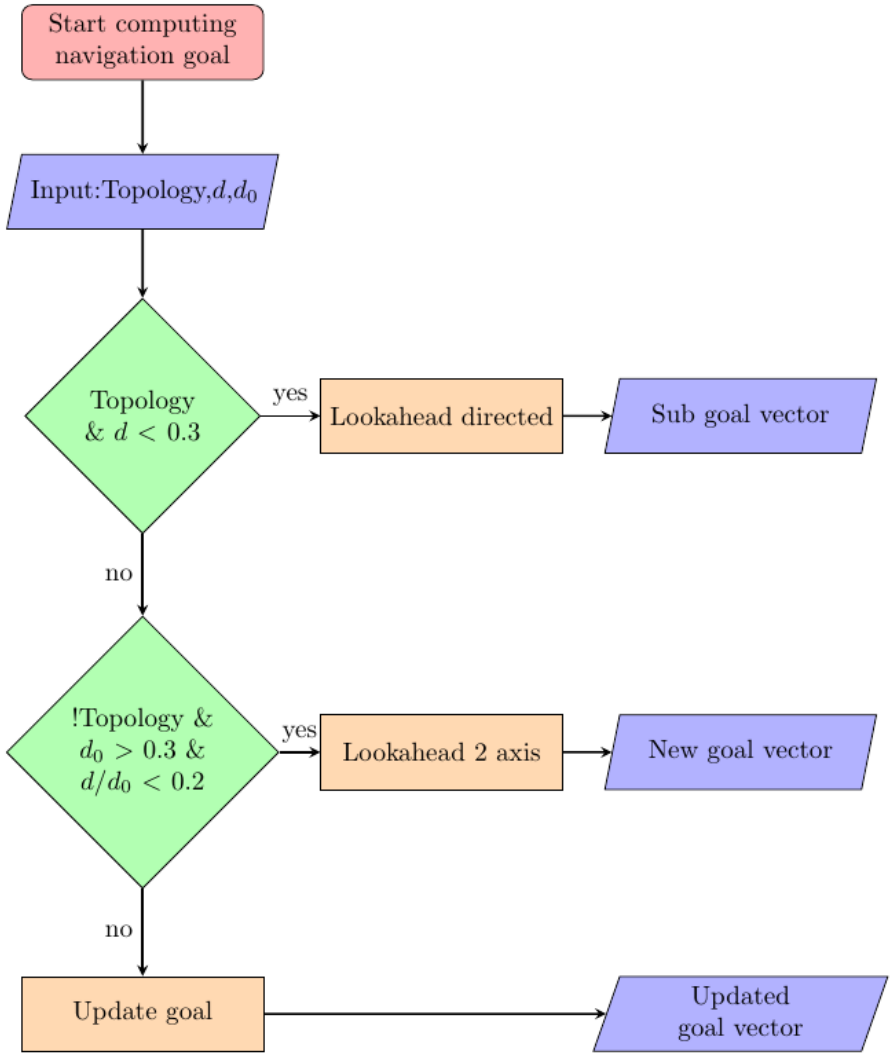

Figure 5.11: Flow chart for the determining the goal vector in navigation phase controller. Toplogy is a Boolean, d is the estimated distance to the next goal and d_0 is the original goal vector.

subsections
It should be noted that the detection result is used for both searching sub-goals and creating the firing map of the BC model. More sensors can increase the accuracy of the firing map but require more computational power. To perform the directed lookahead algorithm, we need to consider the four favorable directions [W, N, S, E], so the number of sensors must be a multiple of four. After considering the trade-off between computational power and the accuracy of the result, we decide to use 16 sensors for obstacle detection.

Vector-based Navigation

During vector-based navigation, the goal is to find the highest reward even if the directions to search for it are unknown. To find the new goal vector, we virtually perform linear lookahead in the x and y directions, meaning that the path taken by the agent is not the one being traversed in reality. The agent's speed is $|v| = 0.5m/s$, and the velocity vector in each of the four lookahead directions is $\overrightarrow{v} \in |v|[\hat{e}_x, -\hat{e}_x, \hat{e}_y, -\hat{e}_y]$. For each direction, we employ the linear lookahead algorithm described in Algorithm 3 [Eng21]. After resetting the GC spiking

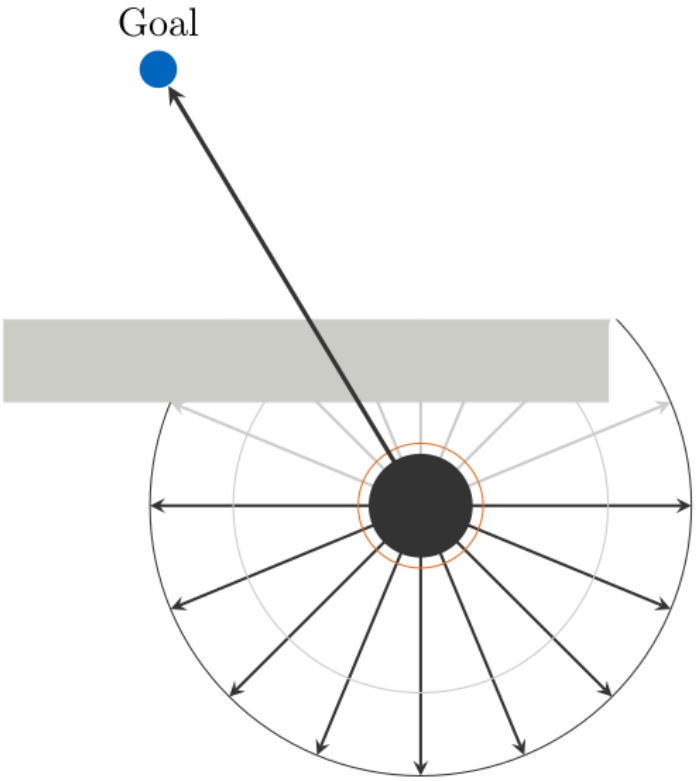

Figure 5.12: Obstacles detection in 16 directions,as well as heading direction and goal vector direction. The directions are defined as blocked if the $d_\theta < 1.3m$ or it's neighboring direction $d_{\theta\pm} < 0.9m$. The goal vector direction is shown in thick arrow and blocked. The light gray arrows represent blocked directions and dark arrows are unblocked directions. The outside dark circel represents the distance $d = 1.3m$. The light grey circle has the distance $d = 0.9m$ and the smallest orange circle marks $d = 0.3m$.

and defining the velocity vector, we perform an iteration with a given number of steps k in each velocity direction. The maximal iteration steps k in each velocity direction depend on the maximal size of the environment since the traveled distance after k steps will exceed the maximal size of the environment. The first step in the iteration is to compute the projected PC firing on the x or y axis, denoted as p_x or p_y, respectively. The GC spiking is projected firstly on the axis and filtered by the connection from GC to PC. A higher spiking value is obtained if both projections align. To implement this, we use a threshold to transform the GC spiking and the connections to a binary matrix. p_x or p_y can be determined by multiplying the two binary matrices and normalizing the result, as shown in Equation 5.40 [Eng21].

$$p_{i|m|x} = (\sum_{x=0}^{n} \hat{S}_{m|xy} * \sum_{x=0}^{n} C_{i|m|xy})/|\hat{s}_m|^2 \tag{5.40}$$

where s_m is the GC spiking vector and S_m is the transformed nxn matrix.The transformed binary matrix $\hat{S}_m$ has entries of 1 if the spiking value $s_{m|j}$ is greater than 0.1, and 0 otherwise. The connection $C_{i|m}$ refers to the connection from GC module m to the place cell i. Using the PC firing values, we can compute the reward firing by multiplying the PC firing with its corresponding reward value in the x or y axis, and then finding the maximal reward spiking using Equation 5.41 [Eng21].

$$b_x = max\{p_x * r\} \tag{5.41}$$

where r is the reward cell and b_x is the reward spiking in x axis. Once the maximal reward spiking for the y-axis is calculated using the same equation as the x-axis, if no previous

Algorithm 3: Linear lookahead along 2 axes

input :
GC network
PC network
Cognitive map
output:
$\overrightarrow{g}$:Goal vector
Reset virtual GC spiking to current spiking s
Define the lookahead velocity vector $\overrightarrow{v} = |v|[\hat{e}_x, -\hat{e}_x, \hat{e}_y, -\hat{e}_y]$
for each $\overrightarrow{v}$ **do**
  for k in max number of steps **do**
    Based on s, compute projected PC firing $p_x or p_y$;
    Identify the place cell with max reward firing $b = max\{p * r\}$;
    if no $b_{max}^{0...k-1}$ for that axis exists or $b^k > b_{max}^{0...k-1}$ **then**
      Save direction,reward $b^k$and distance d traveled
    end
    if $b_{max}^{0...k-1}$ exists and k>50 and $b^k < 0.85 * b_{max}^{0...k-1}$ **then**
      break
    end
    Apply $\overrightarrow{v}$ to the virtual GC spiking
  end
  Reset virtual GC spiking to current spiking s
end
Consolidate goal vector as $\overrightarrow{g} = [d_x, d_y]$;
return Goal vector $\overrightarrow{g}$

reward spiking was found in that axis or a new reward spiking exceeds the previous values, the direction, reward spiking, and the traveled distance are saved. The distance traveled is calculated using the formula $d = |v| * dt * k$. If a previous reward spiking existed but falls below a threshold of $b^k < 0.85 * b_{max}^{0...k-1}$ after $k = 50$ steps, the lookahead in that axis is stopped.
After that, the velocity vector is applied to the virtual GC and the GC spiking is updated. Then, the iteration is started again to compute the directed PC firing and maximal reward spiking for each axis. It is not necessary to evaluate the PC and GC firing at every time step, so the values are evaluated at every $n = 10_{th}$ time step of the GC. When one direction is completed, the GC spiking is reset to the current spiking and the iteration starts on the next axis.
Once the iterations are completed for all four directions, the goal vector $\overrightarrow{g} = [d_x, d_y]$ is determined using d_x and d_y, which are the traveled distances to the highest reward spiking in the corresponding axis. This goal vector is then applied to the agent, and the agent starts to move towards the goal.

Topology-based Navigation

If the agent encounters obstacles that block its path to the goal vector or heading direction ($d < 0.6m$), it will switch to topology-based navigation. The cognitive map, which was generated during exploration, is used to determine the optimal sub-goal for the agent. A directed lookahead algorithm is performed in the directions where obstacles are detected. The algo-

rithm only considers the favorable four directions [W,E,S,N] in the "linear sunburst" maze model. The GC is updated with a velocity vector $\overrightarrow{v} = |v|[\cos\theta, \sin\theta]$, where $v = 0.5m/s$ and θ is the valid direction.
To perform directed lookahead in the obstacle detection directions, we use an algorithm similar to the one used in vector-based navigation. At the beginning of each iteration, we compute the PC firing by multiplying the GC firing with the connections. We then calculate the maximal reward spiking value, b, by multiplying the PC firing value with its corresponding reward value. If the maximal reward spiking value b exceeds the previous value, we save the traveled distance $d = |v| * dt * k$, where k is the iteration steps and dt is the evaluation time step, which is evaluated every $n = 40_{th}$ time steps of GC. We apply the velocity vector to the GC and start the next iteration. To reduce the computational power, we only evaluate the reward spiking at each $n = 40_{th}$ time step.
Once the reward spiking $b_{max}^{0...k-1} > 0.8$ and begins to decrease, the lookahead is stopped when the reward $b^k < 0.85 * b_{max}^{0...k-1}$ in that direction. To save on computational resources, the lookahead is also halted after 50 steps, which corresponds to the agent traversing half of the environment.
Once the lookahead is completed in all directions, the GC firing is reset and the algorithm chooses the direction with the highest reward spiking value b as the most promising travel direction. The agent then moves in that direction until the sub-goal is reached or until other directions become free. In either case, the agent will perform topology-based navigation for sub-goal localization. If all directions become unblocked with distances $d_\theta > 1.3m$ or the agent reaches the goal with a reward value $b > 0.9$, the agent switches to vector-based navigation.

5.8.2 Movement of Agent

Once the agent has computed the goal vector, it applies the velocity vector and starts moving towards the goal. As the agent moves, it uses 16 sensors to detect obstacles in its path, as explained in Section 5.8.1. The sensor data is used to create a firing map of BC, and the agent's speed is then used to update the GC spiking. The updated GC spiking value is then used to compute the PC firing value, and the information in the cognitive map is updated accordingly, as described in Section 5.6. This loop of actions continues until the agent either reaches the goal or encounters an obstacle that requires it to switch to topology-based navigation, as explained earlier.

5.8.3 HDC Model

To summarize, after the agent moves, it computes the changes in its orientation and uses this information to update the stimulus for shift layers in the visual system. Using the position of visual landmarks and the agent's orientation, the ACD and ECD can be computed and sent to the position encoder to check if the cues are in the field of view of the agent. The corresponding stimulus is then set to the rings of ACD and ECD, which are used to update the firing rates of HDC neurons. The head direction, ACD, and ECD can be estimated based on the updated firing rates. Note that the firing rates are not updated if the agent moves straight, is out of the range of visual landmarks, or the visual landmarks are not in the field of view of the agent, in order to reduce computational complexity.

Chapter 6

Experiments

In this chapter, we assess our proposed system, beginning with an overview of the simulation environment setup. Next, we initiate the GC model and store the resulting cognitive map following exploration, which is subsequently used for navigation. The estimated head direction, ECD, and ACD are displayed in the head direction and cue direction map. By detecting obstacles, we can generate the BC firing map. During navigation, the agent can switch between topology-based and vector-based methods.

6.1 Setup

The system is developed using Python 3.8, and the simulation environment is the PyBullet physics simulator. The Pioneer P3DX is utilized as the agent, equipped with 16 sensors to detect obstacles in the environment. The neurons in the GC networks have four preferred directions, which are the only directions considered during topology-based navigation. Thus, the environment should enable the agent to move in four specific directions without obstructions. Therefore, a square maze with obstacles for detection is selected. To minimize the exploration time, a predefined list of locations is used to rapidly create the cognitive map. The location list should be easily determined. The environment chosen for implementation is displayed in Figure 6.1. The maze size is 11m x11m, with a visual landmark (light blue) located at $[x, y] = [5, 5]$ and a goal position of $[1.5, 10]$. The agent is equipped with 18 lines, containing 16 sensors for obstacle detection, one head direction, and one goal vector direction. Green lines denote unobstructed directions, while red lines denote blocked directions. The origin of the coordinate system is located at the bottom left, and the agent's initial position is at $[5.5, 0.55]$. The environment contains obstacles (light gray) and a maximum of five doors (dark gray). In Figure 6.1, doors three and five are open, while doors one, two, and four are closed. The minimal velocity scaling factor for the GC model is set to $g_m = 0.2$ to ensure that a specific set of GC fires in each location of the environment. The simulation time is set to $dt = 10ms$, and the agent moves at a constant velocity of $v = 0.5m/s$.

6.2 Grid Cell model

To initialize the system, we apply a velocity of $[0.2, 0.2]m/s$ to the network with a 95% probability. We test the GC firing with different initialization steps, and the results are shown in Figure 6.2. For $k = 2000$ steps, the GC patterns of modules 1 and 3 are similar, while for $k = 4000$ steps, the GC patterns of modules 4 and 5 are similar. To ensure that the lookahead

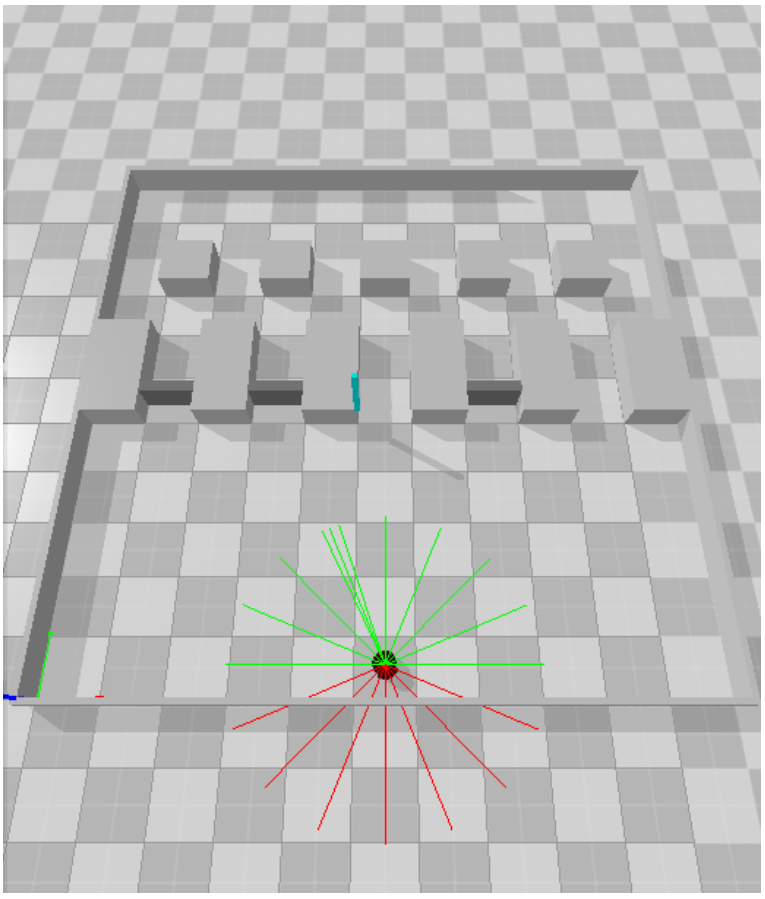

Figure 6.1: Simulation environment. The agent, equipped with sensors, moves at a constant velocity while navigating the environment. The obstacles in the environment are depicted in light gray, while the doors are shown in dark gray. The visual landmark is located at $[5, 5]$, and the goal position is located at $[1.5, 10]$.

works correctly in the favorable directions, we choose the state where the grid orientations are unique in different modules. To minimize computational power, we set $k = 1000$ steps for initialization. Additionally, we tested the effect of the simulation step time on the GC pattern and found that the simulation step dt must be below 20ms, otherwise the regular triangle of GC firing will not be generated.

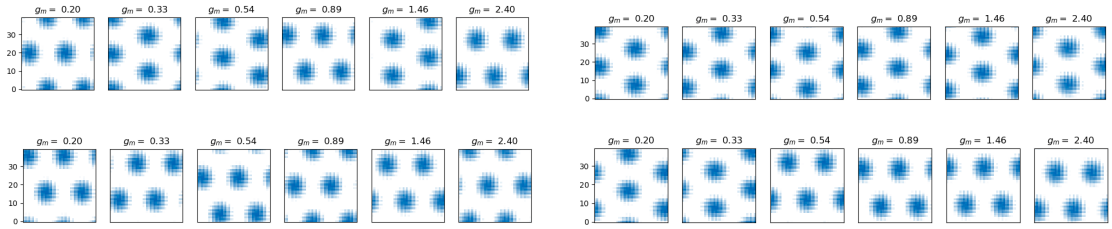

Figure 6.2: Initialization of GC modules after different steps.(Top left)Shift of GC pattern of $k = 1000$ steps. (Top right) GC pattern after $k = 2000$ steps. (Bottom left) GC pattern after $k = 3000$ steps. (Bottom right) GC pattern after $k = 4000$ steps.

6.3 Cognitive Map

Figure 6.3 shows the cognitive map that was obtained after 15000 steps of exploration, which can now be used for navigation. The agent initially moves straight ahead from the position $[5.5, 0.55]$ and explores the left side of the maze before moving towards the right side and passing through door five. Note that doors in the environment can be either closed or open. After reaching the goal at $[1.5, 10]$ (indicated by the blue dot), the agent continues to explore the remaining areas around the small obstacles to complete the cognitive map. The visual landmark is located at $[5, 5]$ and is indicated by the orange dot.

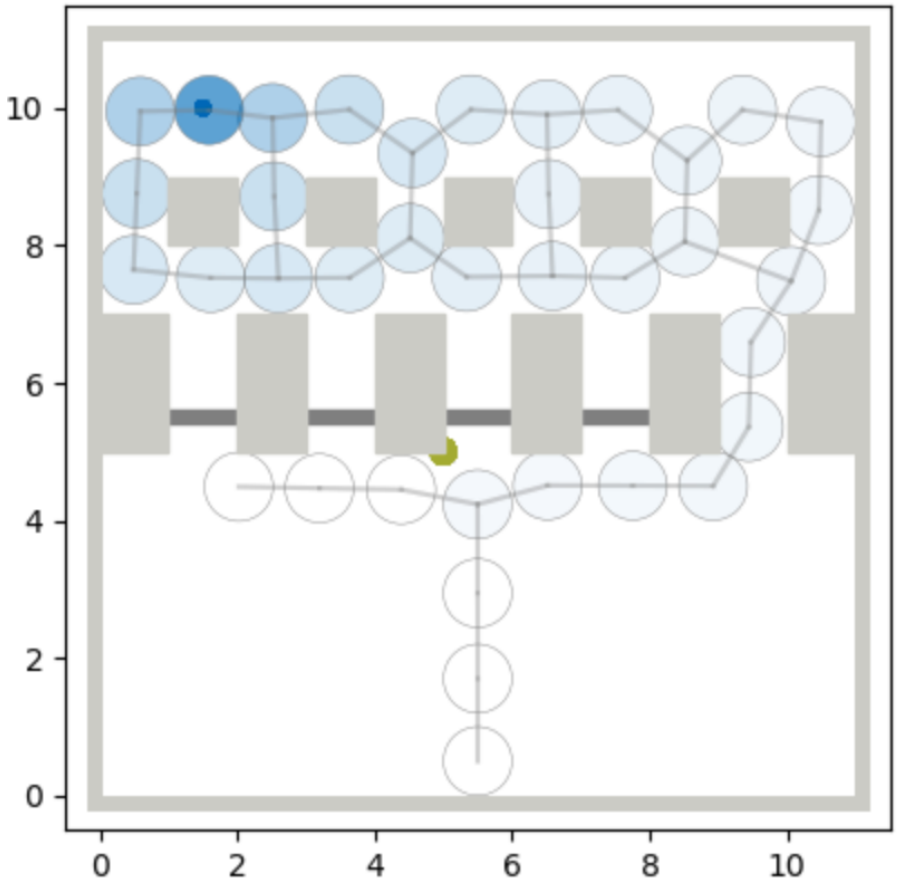

Figure 6.3: Cognitive map after exploration of 15000 steps. Box obstacles are depicted in light gray, while doors are shown in dark gray. The goal is indicated by a blue dot, and the visual landmark is marked by an orange dot. The circles represent place cells and their respective fields, with the color of the place cells being dependent on their reward value. A darker blue color indicates a higher reward value for the place cells.

6.4 Head Direction and Cue Direction Map

In Figure 6.4, we can see the ACDC and ECDC active when the agent is in the range of the position encoder and the cue is in the field of view. The position encoder has a range of 4m x 4m with the cue at the middle point. The agent's orientation is 110.7°, and it is located approximately at $[4.5, 3.5]$, with a distance vector to the cue of $[1, 2]$, which is within the range of the position encoder.

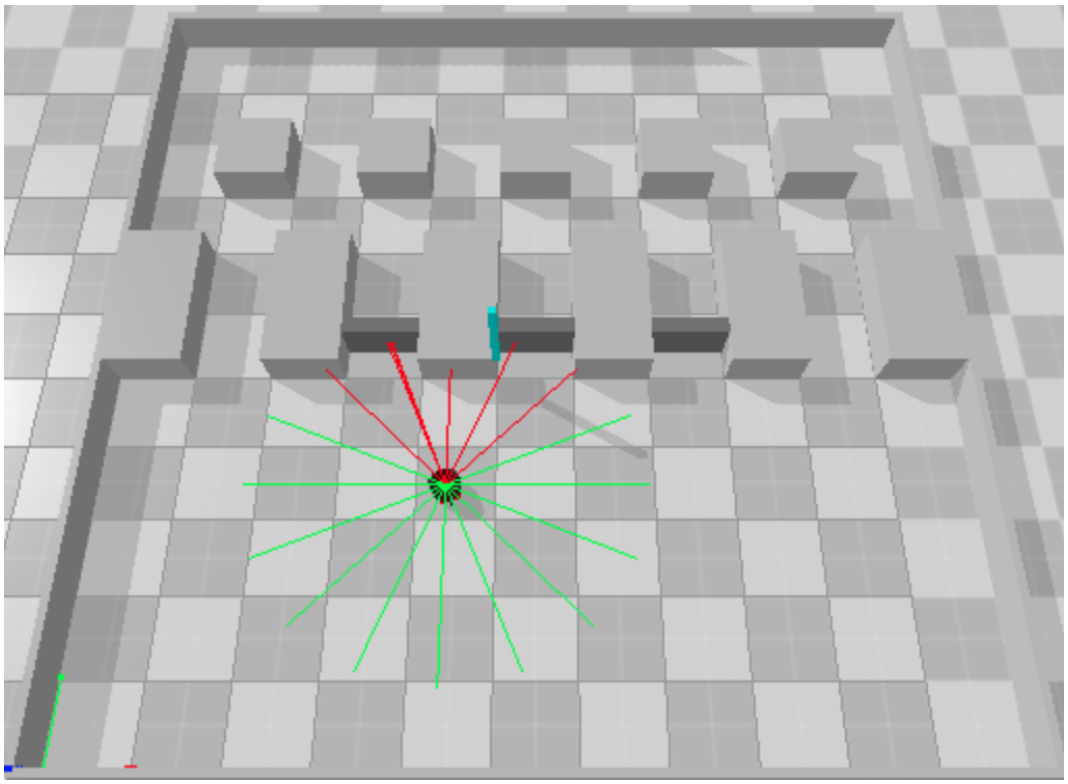

Figure 6.4: When the agent enters the range of the landmark and the visual landmark is in the field of view of the agent, the ACDC and ECDC are activated.

Based on the Figure 6.5, it can be observed that the head direction signal is decoded into allocentric and egocentric representations using the rates of corresponding rings.
The agent's real head direction is 110.7° and after decoding, the allocentric head direction is also 110.7° due to the calibration using the cue information. The egocentric head direction representation shows the agent's viewpoint relative to the cue. The cue is located in the front and right side of the agent in the egocentric frame, which is also consistent with the cue location shown in Figure 6.4. The decoded allocentric head direction signal also shows that the cue is located at 67.5° in the allocentric frame.
The position encoder is used to calibrate the estimated head direction using the visual landmark. The calibration process during exploration is shown in Figure 6.6.
At the beginning of the exploration, the agent moves straight until 8 seconds and the error between the real and estimated head direction is 0°. After 8 seconds, the agent turns left, and the error increases initially and then reduces to −5° when it leaves the range of the position encoder at 10 seconds. When the agent enters the range of the position encoder again at 20 seconds and moves right straight, the error is reduced and almost kept near 0° due to the calibration using the visual landmark.
After that, at 24 seconds, the cue is not in the field of view of the agent, and the error slightly increases. At each turning, the error arises and accumulates due to path integration. After reaching the goal at 90 seconds, the agent continues to explore to complete the cognitive map.

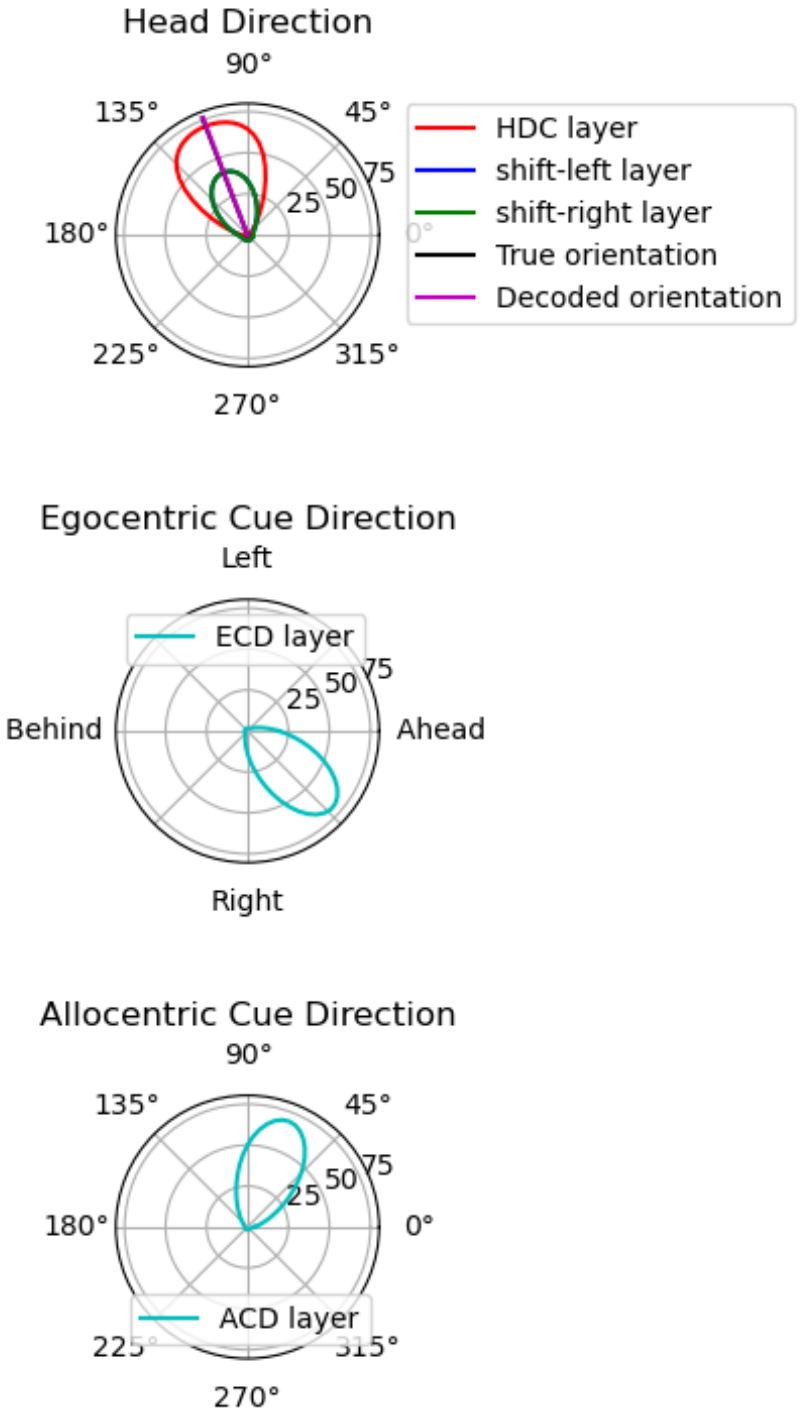

Figure 6.5: The decoded Head direction(top), egocentric cue direction(middle) and allocentric cue direction(bottom).

6.5 Receptive Fields of Boundary Cell Model

When the agent detects obstacles, as shown in Figure 6.4, the BC model fires and generates a receptive field, which is shown in Figure 6.7. The firing map of the BC model is generated

based on the obstacles detected by the agent. The size of the receptive field depends on the distance between the agent and the obstacles, and the firing rates are higher when the agent is closer to the obstacles. The accuracy of the firing map can be improved by using more sensors, but this would require more computational power for obstacle detection.

6.6 Navigation

After saving the cognitive map during the exploration phase, the agent proceeds to use it for the navigation task. Initially, the agent performs navigation in the same configuration as the exploration phase to ensure that the cognitive map is functioning correctly. Afterward, we change the configuration of the maze and test the agent's performance with the same cognitive map that was saved during the exploration phase.

6.6.1 Goal and Sub-goal Localization

The agent navigates in a maze with the same configuration as in the exploration phase, but with only door five open. It begins by using vector-based navigation to move towards the goal. When obstacles block its path, the agent switches to topology-based navigation to determine the direction with the highest reward spiking. In topology-based navigation, the agent only considers the preferred directions of North, South, East, and West. Once all directions become free or the agent is close to the goal, it switches back to vector-based navigation. To reduce computational demands, spiking is only checked every 10 time steps for linear lookahead along two axes and every 40 time steps for directed linear lookahead. Figure 6.8 illustrates goal and sub-goal localization at various time steps.

6.6.2 Navigation in Different Maze Configurations

In the navigation tasks performed in different maze configurations (Figure 6.9), the agent encountered different obstacles and had to adjust its path accordingly. When doors three and five were opened (bottom left of Figure 6.9), the agent arrived at door two along the vector between its initialization location and the final goal position (dark arrow in top left of Figure 6.8). The agent then had to decide whether to move west (an invalid path due to obstacles encountered during exploration) or east (a known valid path). The agent moved towards the east and found a sub-goal when it passed by door three, then moved north towards the sub-goal before continuing along the path it had taken when only door five was open.
A similar situation occurred when door two was opened or when all doors were open, with the agent taking the path through door two and finding a sub-goal using directed lookahead. Despite the fact that the path through door one was the shortest in the configuration, the agent never considered it and instead took other doors. Consequently, the agent became stuck and could not reach the goal if only door one was open. Throughout the navigation in various maze configurations, we also measured the difference between the estimated and actual head direction. The results are presented in Table 6.1. The smallest error occurred when the agent took the path through door two. In this case, the agent was able to find a sub-goal directly after arriving in front of door two, without turning east. On average, the maximal relative error for estimating the head direction was 2.4%. To test the HDC model's robustness to noise, we applied variable relative noise to the angular velocity, with deviations set at 0.1, 0.2, 0.3, and 0.4. We then opened only door five in the maze and observed the

Open doors	Maximal HD Error	Average HD Error	Maximal Relative Error
Door five	7.6°	2.9°	2.7%
Door one and five	7.6°	2.9°	2.7%
Door three and five	8°	2.7°	2.7%
All doors	3.6°	0.7°	1.4%
Average	6.7°	2.3°	2.4%

Table 6.1: Result of Head direction estimation during navigation in different maze configurations.

agent's ability to find the goal. The agent successfully found the goal when the relative deviations were 0.1, 0.2, and 0.3, but got stuck when the relative deviation of the noise was 0.4. The results are summarized in Table 6.2.

Relative deviation	Maximal HD Error	Average HD Error	Maximal Relative Error
0.1	6.4°	2.3°	2.7%
0.2	10.2°	4.6°	2.9%
0.3	11.9°	5.6°	4.2%
0.4	23°	6.7°	5.1%

Table 6.2: Result of HD error with noise of different relative deviation.

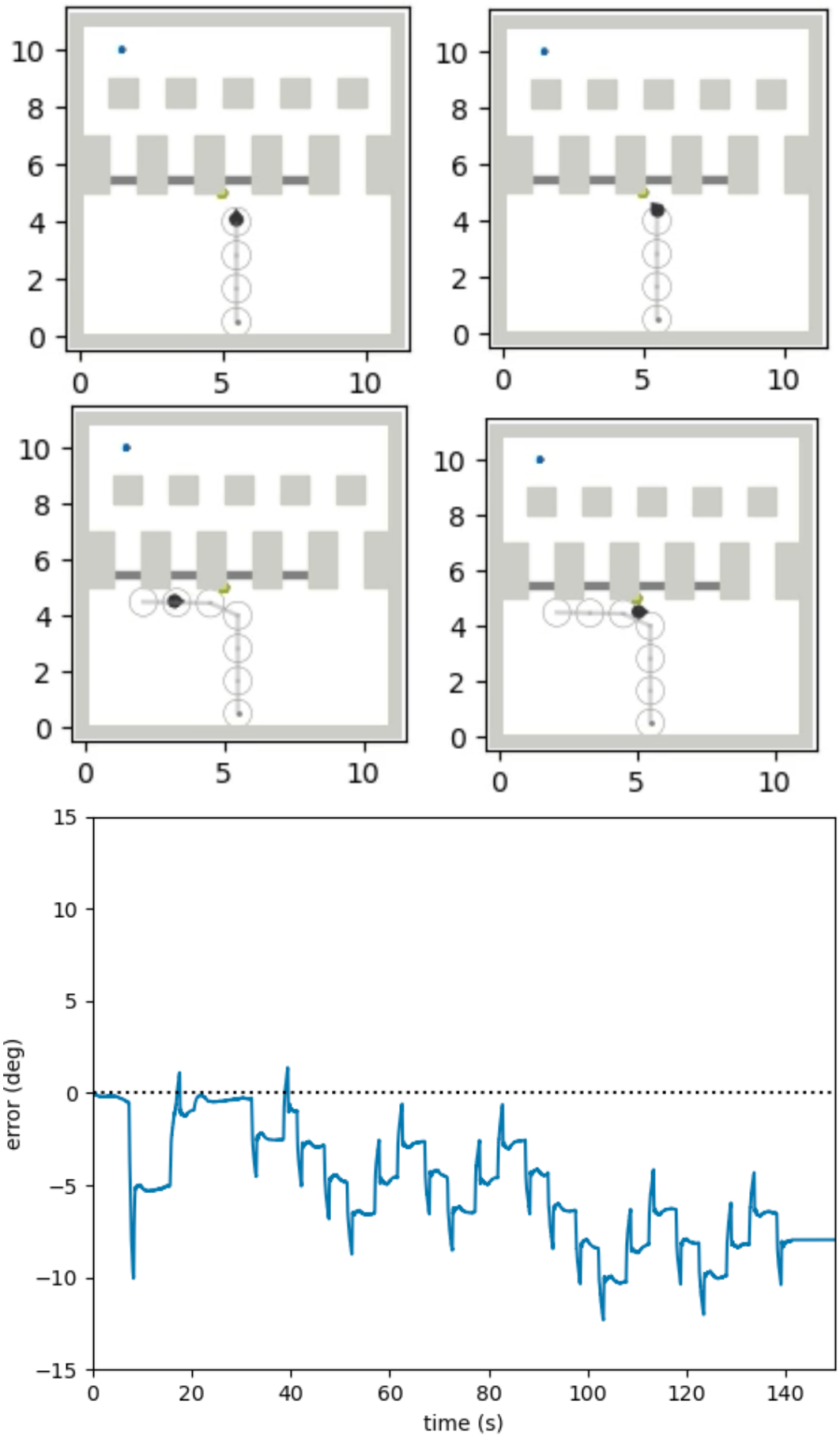

Figure 6.6: Head direction calibration using position encoder. (Top left) The agent moves straight ahead and enters the range of the position encoder until 8 seconds. (Top right) The agent begins to turn and leaves the range of the position encoder after 8 seconds. (Middle left) The agent returns and re-enters the range of the position encoder, and the estimated head direction is calibrated at 20 seconds. (Middle right) The agent leaves the range of the position encoder after 24 seconds. (Bottom) The error between the real and estimated head direction changes during the calibration process.

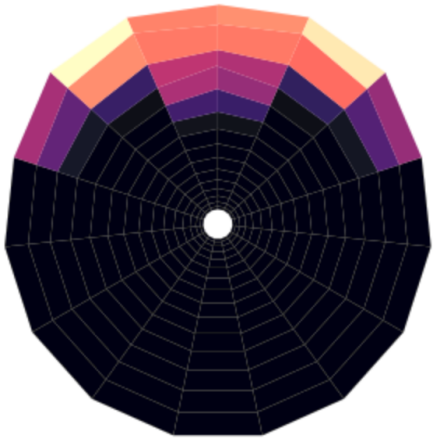

Figure 6.7: The receptive field of Boundary cell model in egocentric frame. The orientation of the agent is 110.7°.

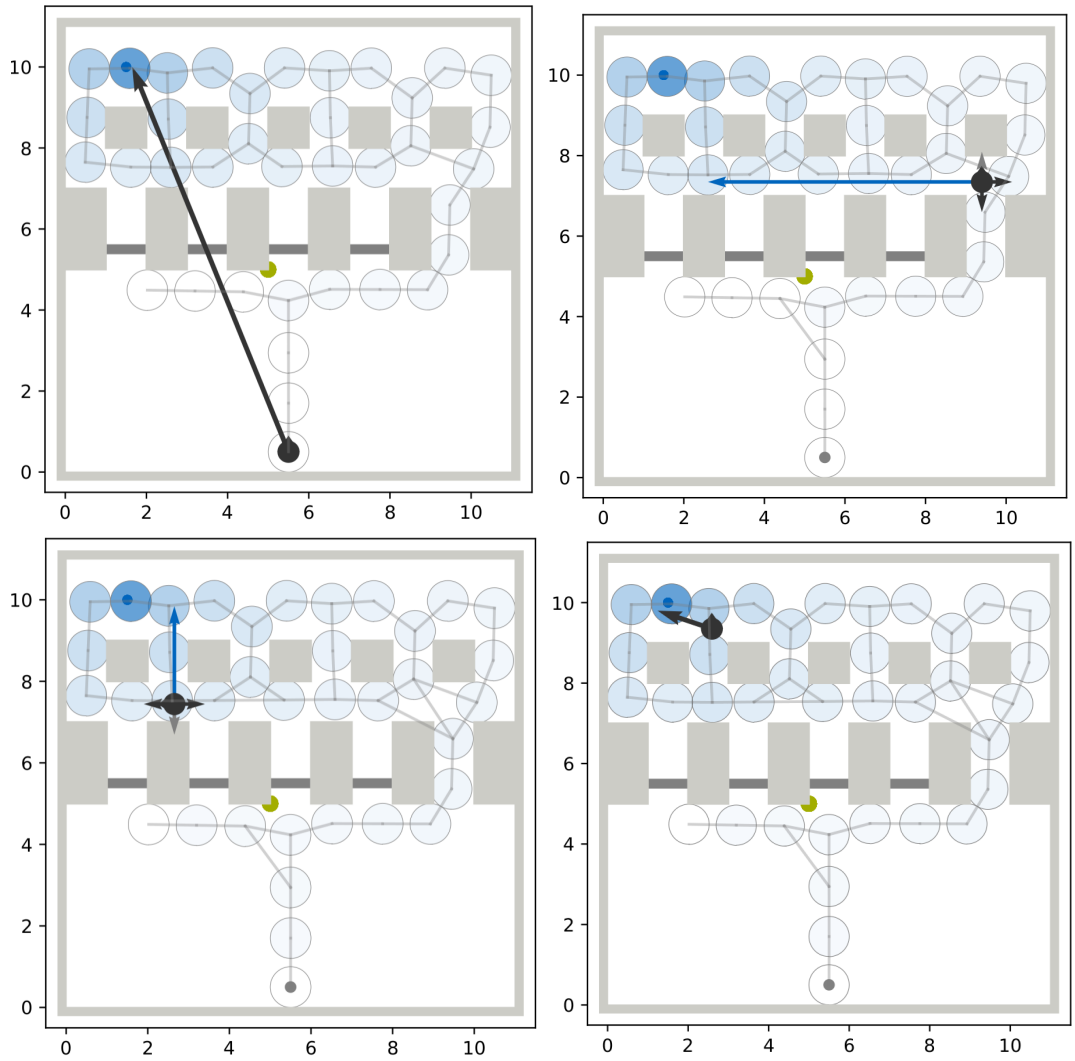

Figure 6.8: Vector-based (in dark arrow) and topology-based navigation(blue arrow) in same maze. The blocked directions are in grey. (Top left) The goal vector is shown for goal localization using linear lookahead along two axes (Algorithm 3) at time step $k = 0$. The blocked directions are in gray. (Top right) Sub-goal localization at time step $k = 3538$ is shown using directed linear lookahead to find the direction with the highest reward spiking (blue arrow). (Bottom left) Sub-goal localization at time step $k = 4909$ is shown. (Bottom right) The agent switches to vector-based navigation when it is close to the goal.

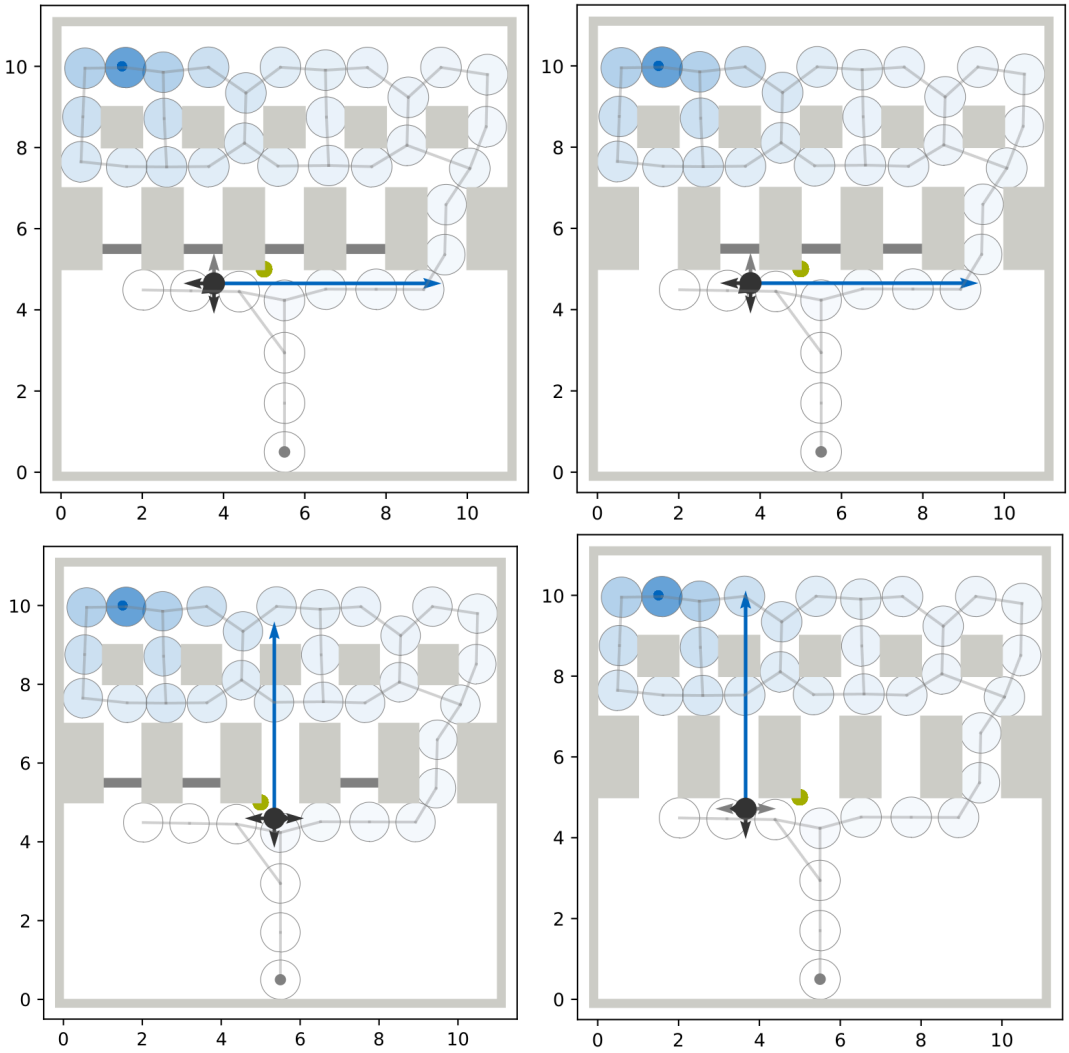

Figure 6.9: Navigaiton in different maze configurations. (Top left) Only door five is open, and the agent successfully navigates to the goal. (Top right) Both doors one and five are open, and the agent still chooses to go through door five. (Bottom left) Doors three and five are open, and the agent takes the shortcut through door three. (Bottom right) All doors are open, and the agent chooses to go through door two.

Chapter 7

Discussion

In this section, we will assess the effectiveness of our proposed system. Firstly, we will examine the strengths and characteristics of the system. Next, we will discuss its limitations. Finally, we will evaluate the biological plausibility of the system.

7.1 Advantage of the Proposed System

In this study, we integrate several cell models including HDC, PC, GC, BC, and prefrontal cortex cell to provide a comprehensive understanding of their working principles. The system is tested in the Pybullet physics environment where the agent can construct a cognitive map during the exploration phase and use it for topology-based navigation to search for subgoals. The agent can switch to vector-based navigation when facing obstacles and ultimately reach the goal. Moreover, when other doors are open, the agent can find shortcuts by using directed lookahead. Additionally, the head direction can be estimated and calibrated with a visual landmark, resulting in a maximal relative error of 2.7%, which is acceptable. The firing map of the boundary cell can be generated, indicating the firing information based on the agent's distance from the obstacles. The system only requires minimal inputs such as angular velocity and speed to update the neuronal spiking, making it efficient in exploration and navigation tasks.

7.2 Limits of the Proposed System

7.2.1 Computer Requirements

The first limitation of this system is the high computational effort required. The system includes several basic models, each consisting of many neurons. There are a total of 20,600 neurons in the HDC model and 9,600 neurons in the GC models. In each simulation step, the spiking in these neurons needs to be updated, which requires a high amount of computer power. To mitigate this issue, the spiking of neurons was only updated when the neuron was active. Even with the use of an NVIDIA Geforce RTX 3060 to accelerate computation, the real-time required for exploration was 2.9 hours and for navigation, it was 1.4 hours. This is despite the simulated time being only 150 seconds for exploration and 80 seconds for navigation. The real-time required was almost 60 times that of the simulated time, which made the debugging of the system difficult during the implementation of the system in Python.

7.2.2 Straight Topology-based Navigation

The neurons in the GC model have preferred firing directions in the south, east, west, or north. This is similar to the neurons in HDC, which also have preferred firing directions but with only 100 neurons for a whole circle of 360°, resulting in a resolution of 3.6°. The properties of the GC model lead to that the agent only considers these four favorable directions for directed linear lookahead during topology-based navigation. Consequently, the agent cannot move towards other directions except the four favorable directions, which limits its ability to find shortcuts in other non-squared maze environments.

7.3 Biological Plausibility

The proposed system is composed of several basic models. In the absence of visual landmarks, the HDC model can estimate the head direction based on angular velocity with a small error, which is consistent with biological observations showing that animals can find their way in the dark [Tau90]. The firing profile of the HDC model follows a Gaussian curve with a peak firing rate of approximately 70 Hz, which falls within the biologically plausible range [Tau07]. While the shifting function used in the HDC model is not biologically accurate, it can still be used to estimate the head direction during agent turning, and is based on previous work that focused on robotic applications [Sew20].
In the presence of visual landmarks, we proposed a position encoder to calibrate the estimated head direction, simulating the use of landmarks as reference points in animal navigation. However, the position encoder is implemented in an artificial manner, rather than biologically accurate. It stores the matrix of positions and resets the ACD information when the agent re-enters the range of the position encoder. To estimate the relevant information of ACD and ECD, we built a neuron sheet of CCF1 and CCF2. These conjunctive neuron models are also artificial but provide useful cue information for the agent.
The BC model produces a firing map that indicates the relationship between the agent and obstacles. A boundary coding neuron in the model has a receptive field around the agent, and its firing field becomes smaller and its firing rate higher as the agent approaches obstacles, which is consistent with biological observations [Lev+09]. While we omitted the firing map in the allocentric frame to reduce computational demands, the estimated head direction from the HDC model can be used to derive allocentric information for the BC model. However, this aspect was not explored in our work.
We utilized multiple 2D neuron sheets to construct the GC model, combining them in a biologically supported manner as described in a previous work [Edv17b]. The firing pattern of the GC model forms a regular triangle, which is consistent with the properties observed in previous studies [May15]. The cognitive map in our model is based on the firing of neurons in other cells, and the GC firing can propagate through the network of PCs and prefrontal cortex cells. These prefrontal cortex cells are believed to be involved in navigation [Eic17], and the PC has been identified in animal brains [OKe76]. To facilitate the agent's goal-directed behavior, we proposed an algorithm called linear lookahead, which is an artificial but effective method for helping the agent find the sub-goals or goals.
The proposed system is composed of the basic models previously discussed and their implementation is based on previous works such as [Sew20], [Nit21], [Hey21], and [Eng21], which have focused on robotic applications. Combining these models is an assumption based on these previous works and allows the agent to find the goal and provide information about head direction and boundary coding firing with an acceptable level of error.

Chapter 8

Conclusion and Outlook

In this chapter, we will first provide a summary of our work. Then, we will offer suggestions on how to extend our system in future research.

8.1 Conclusion

Our goal was to create a general system that could enable agents to explore and navigate by providing information about HD and boundaries, using existing basic models. We have successfully proposed such a system.
Our work involved integrating several basic models to develop a comprehensive system that facilitates navigation and exploration by providing information on head direction and boundaries. The HD model was implemented using the CANN network with a Gaussian firing profile, which could be shifted by applying angular velocity to the shift layers. We also developed ACD and ECD neuron rings to update cue information in the egocentric and allocentric frames, respectively. The CCF neuron sheet could estimate either HD based on ACD and ECD or ACD based on HD and ECD. Additionally, a position encoder stored the position matrix, which enabled calibration of the estimated head direction when the agent was in the vicinity of a visual landmark. The BC model included a firing map to simulate boundary decoding neurons' properties, such as firing at specific distances and orientations from obstacles. The GC model was developed using a nested approach, combining six modules with 2D neuron sheets, with each neuron having one of the four preferred firing directions. This allowed the agent to search for sub-goals in topology-based navigation. Based on the GC model's firing rate, the PC firing rate could be calculated, and the prefrontal cortex cell models could store the cognitive map's necessary information.
Our proposed system was implemented using Python 3.8 and tested in the PyBullet physics simulation environment. We designed a square maze with various configurations and tested the system in both exploration and navigation phases. During the exploration phase, the agent explored the maze to generate a cognitive map. In the navigation phase, the agent utilized vector-based and topology-based navigation strategies depending on the obstacle distance and orientation. The agent could use the cognitive map to perform linear lookahead algorithms to find sub-goals and the goal, with acceptable head direction error. Furthermore, the agent was able to find shortcuts when the maze configuration changed.
The proposed system is computationally expensive despite our efforts to reduce it by using NVIDIA Geforce RTX 3060 for acceleration. The simulation time is still about 60 times the simulated time, which makes debugging challenging. Navigation in the 80s takes about 1.4 hours, further exacerbating the computational cost. Nevertheless, the proposed system presents a comprehensive working model of the brain, incorporating several models of cell

types already found in animals' brains. This system provides a foundation for future research aimed at developing more accurate models.

8.2 Outlook

Despite our efforts, there are still several ways in which our proposed system can be further extended and improved.

8.2.1 Additional Cell

To enhance the realism of our proposed system, we could consider including a speed cell model, as these cells have been shown to fire linearly with the running speed of animals [Kro15]. Currently, in our system, the agent moves at a constant speed during testing. However, if we want to simulate a more realistic scenario, where the agent moves at variable speeds, a speed cell model would be necessary.

8.2.2 Improvement of Grid Cell Model

While our current GC model only allows the agent to consider four preferred firing directions, there is room for improvement to enhance the agent's ability to find shortcuts. One alternative is to implement a more flexible GC model that enables the agent to move in any direction. For example, Zibo Zhou proposed a GC model consisting of four shift layers and one value space layer that can construct a 3D twisted torus, allowing for more varied movement directions [Zho21]. However, this model is computationally expensive and may require optimization for practical implementation.

8.2.3 Multiple landmarks

To calibrate the estimated head direction (HD), we used a proximal visual landmark located in the middle of the maze, and a position matrix of 4m x 4m around the landmark. However, an alternative approach could be to use a distal cue for calibration, as it moves less rapidly when the agent moves and can be used more frequently during navigation or exploration. Multiple visual landmarks can also be used for calibration instead of only one, such as a combination of distal and proximal visual landmarks. The distal landmark can estimate orientation similar to a compass, while the proximal landmark can provide distance information for finding a sub-goal [KH11].

Appendix A

Appendix 1

A.1 Implementation Details

A.1.1 Configuration of Computer Environment

Our implementation of the proposed system is based on existing code from EI-Sewisy [Sew20], Dominik Nitschke [Nit21], Camillo Heye [Hey21], and Tim Engelmann [Eng21]. The system is implemented in Python 3.8 and tested using Pybullet simulation. For testing, we used a computer with the following specifications:

- Operating System: Ubuntu 20.04.
- Graphics card: Nvidia GeForce RTX3060 (Driver Version: 515.43.04).
- CPU: Intel(R) Core(TM) i5-10400F CPU @ 2.90GHz.
- Memory: 16GB system memory.
- Disk: SSD NVMe 500G.

A.1.2 Hyperparameters of the Models

- Number of neurons in rings of HDC,SSL,SLR,ACD and ECD: 100
- Number of neurons in CCF1 or CCF2 ring: 100 x 100
- Number of neurons pro GC sheet: 40 x 40
- Number of GC modules in GC model: M=6
- Number of sensors for obstacles detections: 16
- Steps for exploration: 15000
- Steps for navigation: 8000
- Time step dt for simulation and GC firing updating: 1e-2s
- Time step for HDC firing updating: 0.0001s
- Location of goal in maze: [1.5,10]
- Location of visual landmark in maze: [5,5]

List of Figures

2.1 Standard description of connectivity and neural representation of space within the rodent hippocampal formation[Nil22]. The rodent hippocampal formation is interconnected through projections from the medial and lateral entorhinal cortex (mEC and lEC) to CA3 and CA1 via the dentate gyrus (DG). CA3 also sends projections to other networks. DG, mEC, and lEC project to CA3, CA1, and subiculum (SUB), with CA1 projecting back to mEC/lEC through SUB, and SUB also projecting directly to mEC/lEC via presubiculum (Prs) and parasubiculum (Pas). This diagram does not include all projections. The illustration of place cells (PCs), boundary cells (BCs), head direction cells (HDCs), and grid cells (GCs) shows a rodent's trajectory with spiking activity depicted as a heatmap, where warmer colors indicate increased activity. 6

2.2 Grid cells and place cells. The left image depicts a GC from the entorhinal cortex of a rat brain, showing the black trace of the rat's trajectory and red dots representing spikes in a maze with a 1.5m diameter square. The regular triangles are formed by connecting the distribution of the spikes. The right image shows both GC and PC, with the top panel depicting the spike trajectory and the bottom panel displaying a color-coded rate map where red indicates high activity and blue indicates low activity [May15]. 6

2.3 Firing rate vs. head direction plot to characterize the firing properties of head direction cells [Tau09]. 7

2.4 Two different receptive field.(Left) When a rat is close to a boundary, the receptive field of the boundary cell is smaller. (Right) when the rat is far away from the obstacle, the receptive field of the boundary cell is larger [BB20] [Hey21]. 8

2.5 (Top) A firing rate map of a boundary vector cell, which shows the cell's firing rate in different locations of the environment. (Middle) An illustration of the receptive field of the cell at a specific distance and orientation from the rat to the obstacle. (Bottom) A vector representing the direction from the rat to the receptive field of the cell. These figures provide a visual representation of how boundary vector cells respond to the presence of boundaries in the environment [Hey21] [BB20]. 8

3.1 Reference frames: the egocentric frame in orange is body-centered representation, the allocentric in blue is world-centered representation [Nit21]. 9

3.2 Schematic of HDC Attractor.(Left) Three curves of neurons with different Preferred Firing Direction(PFD) (-90°,-22.5° and 45°) in the HD network,the HD is 90°. The activity profile of the network is shown as a black dashed line, which maps the PFD of the HDC network. The activity profile looks similar to the tuning curve of an individual neuron with PFD equal to HD [Sew20]. (Right) Illustration of HDC Attractor and shift layer. The figure depicts an HDC Attractor Network at the inner circle, and a shift right layer at the outer circle (the shift left layer is not shown as it is a mirror of the shift right layer). Each neuron in the shift layer is connected to all neurons in the HDC network. A stable peak in activity is achieved through local excitation (arrowhead) and global inhibition (solid circles) by carefully chosen weights. An external stimulus applied to the shift layer can shift the activity profile in the HDC network [Sew20]. 11

3.3 Discretization of linear path using combination of modules with a phase offset.(Top) A single GC model with a scale of λ is shown to fire periodically at several locations. (Bottom) Combination of multiple GC modules with the same scale but different phase offsets, denoted by ϕ. 13

3.4 Schematic of HDC Attractor.(Left) Three curves of neurons with different Preferred Firing Direction(PFD) (-90°,-22.5° and 45°) in the HD network,the HD is 90°. 14

3.5 Example of a phase offset detector p_j. The 40 x 40 matrices depict the weighted vectors of GC activity in the current location (s_m) and the desired target location (t_m), which are used as inputs for the phase-offset detectors [Edv15]. . . 16

5.1 Overview of the system. The system utilizes the goal vector to perform obstacle detection, which in turn is used to update the Boundary Cell firing map. The agent's position and angular velocity are used to decode the head direction information. The firing of Place Cells can be computed based on the speed of the agent and the firing of Grid Cells. The relevant information for building the cognitive map is stored in the linked prefrontal cortex cells. This process is repeated in each iteration of exploration or navigation until the agent reaches the goal or the cognitive map is complete. 21

5.2 Detailed structure of prefrontal cortex cell. The computation of the firing rate of place cells is based on the firing of grid cells. Each place cell is connected to a prefrontal cortex cell model that includes three cells: a recency cell, which keeps track of visited places, a topology cell, which encodes the spatial relationships between locations, and a reward cell, which identifies the proximity of the goal [Eng21]. 22

5.3 Structure of HDC model.(1) TThe fundamental component of the HDC model is the basic HD model, which estimates the head direction based on path integration and takes angular velocity as input. (2) The Calibration Circuit (CC) is responsible for correcting any errors in the HD estimation by incorporating cues. When a cue is detected by the Visual Detector, the Angle-Cue Distance (ACD) information is inferred by adding the HD and ECD information. The Position Encoder is then used to reset and correct the ACD information. Finally, the HD information is generated by subtracting the ECD information from the ACD information using a Subtractor component [Nit21]. 23

5.4 Comparison of ECDCs (left) and ACDCs (right) with $n = 8$. (Left)The ECDCs represent the landmark location in the egocentric frame of reference, with the agent's orientation fixed at 0°, while the angular position of the landmark depends on the agent's position and orientation. (Right)The ACDCs represent the landmark location in the allocentric frame of reference, with the east of the agent serving as the reference point at 0°, and the angular position of the landmark depending solely on the agent's position [Nit21]. 27
5.5 Angular position of visual landmark in different frames. The HD of an agent can be inferred based on the information from both ACD and ECD. In addition, ACD information can be calculated based on the other two types of information [Nit21]. 28
5.6 CCF1: Adder(left) and CCF2: Substractor(right) with $n = 36$ [Nit21]. 29
5.7 Illustration of the Position Encoder(PE). The PE stores information about the agent's position and the landmark's vector when the agent first enters the range and the visual landmark is in the agent's field of view. This stored information will be used later to decode the ACD. 30
5.8 The conncenction between rings of neurons. The circles represent 1D ring neurons, specifically the HD neurons. The rectangles represent 2D sheets of neurons, such as CCF1. The lines with one arrow represent unidirectional connections, for example, the connection from ECD to HD. The lines with two arrows represent directional connections, such as the connection between SLL and HD. 31
5.9 Illstration of n x n neuron sheet of Grid Cell Module with $n = 4$. All neurons in the GC module are connected along both the east-west and north-south directions. Each neuron is associated with a preferred firing direction $\hat{e}_{\theta_i}$, which can be in any of the four cardinal directions: north, west, east, or south. 37
5.10 Illustration of detailed Navigation. The goal vector is computed by the navigation phase controller and then sent to the agent. In addition, the results of obstacle detection are used to create a firing map for the Boundary Cell model, and the agent's speed is fed to the Grid Cell model. The orientation, position, and location of landmarks are sent to the HDC model for decoding the Head-Direction (HD), Alocentric-Cue (AC), and Egocentric-Cue (EC) information, respectively. 40
5.11 Flow chart for the determining the goal vector in navigation phase controller. Toplogy is a Boolean, d is the estimated distance to the next goal and d_0 is the original goal vector. 41
5.12 Obstacles detection in 16 directions,as well as heading direction and goal vector direction. The directions are defined as blocked if the $d_\theta < 1.3m$ or it's neighboring direction $d_{\theta\pm} < 0.9m$. The goal vector direction is shown in thick arrow and blocked. The light gray arrows represent blocked directions and dark arrows are unblocked directions. The outside dark circel represents the distance $d = 1.3m$. The light grey circle has the distance $d = 0.9m$ and the smallest orange circle marks $d = 0.3m$. 42

6.1 Simulation environment. The agent, equipped with sensors, moves at a constant velocity while navigating the environment. The obstacles in the environment are depicted in light gray, while the doors are shown in dark gray. The visual landmark is located at $[5, 5]$, and the goal position is located at $[1.5, 10]$. 46
6.2 Initialization of GC modules after different steps.(Top left)Shift of GC pattern of $k = 1000$ steps. (Top right) GC pattern after $k = 2000$ steps. (Bottom left) GC pattern after $k = 3000$ steps. (Bottom right) GC pattern after $k = 4000$ steps. 46

6.3 Cognitive map after exploration of 15000 steps. Box obstacles are depicted in light gray, while doors are shown in dark gray. The goal is indicated by a blue dot, and the visual landmark is marked by an orange dot. The circles represent place cells and their respective fields, with the color of the place cells being dependent on their reward value. A darker blue color indicates a higher reward value for the place cells. 47
6.4 When the agent enters the range of the landmark and the visual landmark is in the field of view of the agent, the ACDC and ECDC are activated. 48
6.5 The decoded Head direction(top), egocentric cue direction(middle) and allocentric cue direction(bottom). 49
6.6 Head direction calibration using position encoder. (Top left) The agent moves straight ahead and enters the range of the position encoder until 8 seconds. (Top right) The agent begins to turn and leaves the range of the position encoder after 8 seconds. (Middle left) The agent returns and re-enters the range of the position encoder, and the estimated head direction is calibrated at 20 seconds. (Middle right) The agent leaves the range of the position encoder after 24 seconds. (Bottom) The error between the real and estimated head direction changes during the calibration process. 52
6.7 The receptive field of Boundary cell model in egocentric frame. The orientation of the agent is 110.7°. 53
6.8 Vector-based (in dark arrow) and topology-based navigation(blue arrow) in same maze. The blocked directions are in grey. (Top left) The goal vector is shown for goal localization using linear lookahead along two axes (Algorithm 3) at time step $k = 0$. The blocked directions are in gray. (Top right) Sub-goal localization at time step $k = 3538$ is shown using directed linear lookahead to find the direction with the highest reward spiking (blue arrow). (Bottom left) Sub-goal localization at time step $k = 4909$ is shown. (Bottom right) The agent switches to vector-based navigation when it is close to the goal. 53
6.9 Navigaiton in different maze configurations. (Top left) Only door five is open, and the agent successfully navigates to the goal. (Top right) Both doors one and five are open, and the agent still chooses to go through door five. (Bottom left) Doors three and five are open, and the agent takes the shortcut through door three. (Bottom right) All doors are open, and the agent chooses to go through door two. 54

List of Tables

5.1 Connection weights in basic HD model 32
5.2 Input current U and target firing profile F for calculating weights in Calibration Circuit 34
5.3 Input currents for HDC model 35

6.1 Result of Head direction estimation during navigation in different maze configurations. 51
6.2 Result of HD error with noise of different relative deviation. 51

Bibliography

[AG00] Arleo, A. and Gerstner, W. “Modeling Rodent Head-direction Cells and Place Cells for Spatial Learning in Bio-mimetic Robotics”. In: (June 2000). URL: http://www.aging-vision-action.fr/ava_biblio/pdf/Arleo2000a.pdf.

[Bar+06] Barry, C., Lever, C., Hayman, R., Hartley, T., Burton, S., O’Keefe, J., Jeffery, K., and Burgess, N. “The Boundary Vector Cell Model of Place Cell Firing and Spatial Memory”. In: *Reviews in the neurosciences* 17 (Feb. 2006), pp. 71–97. DOI: 10.1515/REVNEURO.2006.17.1-2.71.

[BB16] Bicanski, A. and Burgess, N. “Environmental Anchoring of Head Direction in a Computational Model of Retrosplenial Cortex”. In: *Journal of Neuroscience* 36.46 (2016), pp. 11601–11618. ISSN: 0270-6474. DOI: 10.1523/JNEUROSCI.0516-16.2016. eprint: https://www.jneurosci.org/content/36/46/11601.full.pdf. URL: https://www.jneurosci.org/content/36/46/11601.

[BB18] Bicanski, A. and Burgess, N. “A neural-level model of spatial memory and imagery”. In: *eLife* 7 (Sept. 2018). Ed. by Colgin, L. and Frank, M. J., e33752. ISSN: 2050-084X. DOI: 10.7554/eLife.33752. URL: https://doi.org/10.7554/eLife.33752.

[BB20] Bicanski, A. and Burgess, N. “Neuronal vector coding in spatial cognition”. In: *Nature Reviews Neuroscience* 21 (2020), pp. 453–470.

[Boc+10] Boccara, C., Sargolini, F., Thoresen, V., Solstad, T., Witter, M., Moser, E., and Moser, M.-B. “Grid cells in pre- and parasubiculum”. In: *Nature neuroscience* 13 (Aug. 2010), pp. 987–94. DOI: 10.1038/nn.2602.

[Bus15] Bush D Barry C, M. D. “sing Grid Cells for Navigation”. In: *The Journal of neuroscience* 87.3 (2015), pp. 507–20. DOI: 10.1016/j.neuron. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4534384/pdf/main.pdf.

[Col+17] Colombo, D., Serino, S., Tuena, C., Pedroli, E., Dakanalis, A., Cipresso, P., and Riva, G. “Egocentric and allocentric spatial reference frames in aging: A systematic review”. In: *Neuroscience and Biobehavioral Reviews* 80 (2017), pp. 605–621. ISSN: 0149-7634. DOI: https://doi.org/10.1016/j.neubiorev.2017.07.012. URL: https://www.sciencedirect.com/science/article/pii/S0149763417301690.

[Edv17a] Edvard I Moser May-Britt Moser, B. L. M. “Spatial representation in the hippocampal formation: a history”. In: *Nature neuroscience* 20.11 (2017), pp. 1448–1464. ISSN: 1097-6256. DOI: https://doi.org/10.1038/nn.4653. URL: https://escholarship.org/uc/item/4w36z6rj.

[Edv20] Edvardsen Vegard, B. “Navigating with grid and place cells in cluttered environments”. In: *Hippocampus* 30.3 (2020), pp. 220–232. DOI: https://doi.org/10.1002/hipo.23147. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/hipo.23147. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/hipo.23147.

[Edv15] *A Passive Mechanism for Goal-Directed Navigation using Grid Cells*. Vol. ECAL 2015: the 13th European Conference on Artificial Life. ALIFE 2022: The 2022 Conference on Artificial Life. July 2015, pp. 191–198. DOI: 10.1162/978-0-262-33027-5-ch039. eprint: https://direct.mit.edu/isal/proceedings-pdf/ecal2015/27/191/1903839/978-0-262-33027-5-ch039.pdf. URL: https://doi.org/10.1162/978-0-262-33027-5-ch039.

[Edv17b] Edvardsen, V. "Long-range navigation by path integration and decoding of grid cells in a neural network". In: *2017 International Joint Conference on Neural Networks (IJCNN)*. 2017, pp. 4348–4355. DOI: 10.1109/IJCNN.2017.7966406. URL: https://ntnuopen.ntnu.no/ntnu-xmlui/bitstream/handle/11250/2479287/postprint.pdf?sequence=1.

[Eic17] Eichenbaum, H. "Memory: Organization and Control". In: *Annual Review of Psychology* 68.1 (2017). PMID: 27687117, pp. 19–45. DOI: 10.1146/annurev-psych-010416-044131. eprint: https://doi.org/10.1146/annurev-psych-010416-044131. URL: https://doi.org/10.1146/annurev-psych-010416-044131.

[Eng21] Engelmann, T. *Biologically inspired spatial navigation using vector-based and topology-based path planning*. Munich, Sept. 2021.

[EH12] Erdem, U. and Hasselmo, M. "A goal-directed spatial navigation model using forward planning based on grid cells". In: *The European journal of neuroscience* 35 (Mar. 2012), pp. 916–31. DOI: 10.1111/j.1460-9568.2012.08015.x. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3564559/pdf/nihms-346909.pdf.

[FBB08] Fiete, I. R., Burak, Y., and Brookings, T. "What Grid Cells Convey about Rat Location". In: *Journal of Neuroscience* 28.27 (2008), pp. 6858–6871. ISSN: 0270-6474. DOI: 10.1523/JNEUROSCI.5684-07.2008. eprint: https://www.jneurosci.org/content/28/27/6858.full.pdf. URL: https://www.jneurosci.org/content/28/27/6858.

[Haf+05] Hafting, T., Fyhn, M., Molden, S., Moser, M.-B., and Moser, E. "Microstructure of a spatial map in the entorhinal cortex". In: *Nature* 436 (Sept. 2005), pp. 801–6. DOI: 10.1038/nature03721. URL: file:///home/he/Downloads/08.09-GridCells2.pdf.

[Har+00] Hartley, T., Burgess, N., Lever, C., Cacucci, F., and O'Keefe, J. "Modeling Place Fields in Terms of the Cortical Inputs to the Hippocampus". In: *Hippocampus* 10 (Jan. 2000), pp. 369–79. DOI: 10.1002/1098-1063(2000)10:4<369::AID-HIPO3>3.0.CO;2-0. URL: https://www.researchgate.net/publication/12337572_Modeling_Place_Fields_in_Terms_of_the_Cortical_Inputs_to_the_Hippocampus.

[Hey21] Heye, C. *Biologically plausible spatial navigation based on border cells*. Munich, Aug. 2021.

[JPS16] Jeffery, K. J., Page, H. J. I., and Stringer, S. M. "Optimal cue combination and landmark-stability learning in the head direction system". In: *The Journal of Physiology* 594.22 (2016), pp. 6527–6534. DOI: https://doi.org/10.1113/JP272945. eprint: https://physoc.onlinelibrary.wiley.com/doi/pdf/10.1113/JP272945. URL: https://physoc.onlinelibrary.wiley.com/doi/abs/10.1113/JP272945.

[KH11] Knierim, J. J. and Hamilton, D. A. "Framing spatial cognition: neural representations of proximal and distal frames of reference and their roles in navigation." In: *Physiological reviews* 91 4 (2011), pp. 1245–79.

[Kro15] Kropff E Carmichael JE, M. M. "Speed cells in the medial entorhinal cortex". In: *Nature* 523.7561 (2015), pp. 419–24. DOI: 10.1038/nature14622.

[Lev+09] Lever, C., Burton, S., Jeewajee, A., O'Keefe, J., and Burgess, N. "Boundary Vector Cells in the Subiculum of the Hippocampal Formation". In: *Journal of Neuroscience* 29.31 (2009), pp. 9771–9777. ISSN: 0270-6474. DOI: 10.1523/JNEUROSCI.1319-09.2009. eprint: https://www.jneurosci.org/content/29/31/9771.full.pdf. URL: https://www.jneurosci.org/content/29/31/9771.

[Lev09] Lever C Burton S, J. A. "Boundary vector cells in the subiculum of the hippocampal formation". In: *The Journal of neuroscience* 29.31 (2009), pp. 9771–7. DOI: 10.1523/JNEUROSCI.1319-09.2009.

[LS92] Löwel, S. and Singer, W. "Selection of Intrinsic Horizontal Connections in the Visual Cortex by Correlated Neuronal Activity". In: *Science* 255.5041 (1992), pp. 209–212. DOI: 10.1126/science.1372754. eprint: https://www.science.org/doi/pdf/10.1126/science.1372754. URL: https://www.science.org/doi/abs/10.1126/science.1372754.

[May15] May-Britt Moser David C. Rowland, E. I. M. "Place Cells, Grid Cells, and Memory". In: Cold Spring Harbor, 2015. DOI: 10.1101/cshperspect.a021808. URL: https://cshperspectives.cshlp.org/content/7/2/a021808.full.pdf+html.

[Mcn+06] Mcnaughton, B., Battaglia, F., Jensen, O., Moser, E., and Moser, M.-B. "Path integration and the neural basis of the 'Cognitive Map.'" In: *Nature reviews. Neuroscience* 7 (Sept. 2006), pp. 663–78. DOI: 10.1038/nrn1932. URL: https://www.researchgate.net/publication/6928350_Path_integration_and_the_neural_basis_of_the_%27Cognitive_Map%27.

[MWP04] Milford, M., Wyeth, G., and Prasser, D. "RatSLAM: A hippocampal model for simultaneous localization and mapping". In: vol. 1. Jan. 2004, 403–408 Vol.1. ISBN: 0-7803-8232-3. DOI: 10.1109/ROBOT.2004.1307183.

[Mos08] Moser EI Kropff E, M. M. "Place cells, grid cells, and the brain's spatial representation system". In: *Annual review of neuroscience* 31 (2008), pp. 69–89. DOI: 10.1146/annurev.neuro.31.061307.090723.

[Nil22] Nils Nyberg Eleonore Duvelle, C. B. "Spatial goal coding in the hippocampal formation". In: *Neuron* 110.3 (Jan. 2022), pp. 394–422. DOI: https://doi.org/10.1016/j.neuron.2021.12.012. URL: https://www.cell.com/action/showPdf?pii=S0896-6273%2821%2901029-1.

[Nit21] Nitschke, D. "Biologically inspired robotic Mapping,Localization, and Navigation using Spiking Neural Networks". MA thesis. Munich: Technical University of Munich, Aug. 2021.

[OKe76] O'Keefe, J. "Place units in the hippocampus of the freely moving rat". In: *Experimental Neurology* 51.1 (1976), pp. 78–109. ISSN: 0014-4886. DOI: https://doi.org/10.1016/0014-4886(76)90055-8. URL: https://www.sciencedirect.com/science/article/pii/0014488676900558.

[OB96] O'Keefe, J. and Burgess, N. "Geometric determinants of the place fields hippocampal neurons". In: *Nature* 381 (June 1996), pp. 425–8. DOI: 10.1038/381425a0.

[OKe71] O'Keefe J, D. J. "The hippocampus as a spatial map. Preliminary evidence from unit activity in the freely-moving rat". In: *Brain Research* 34.1 (1971), pp. 171–175. ISSN: 0006-8993. DOI: https://doi.org/10.1016/0006-8993(71)90358-1. URL: https://www.sciencedirect.com/science/article/pii/0006899371903581.

[Pag+14] Page, H., Walters, D., Knight, R., Piette, C., Jeffery, K., and Stringer, S. "A theoretical account of cue averaging in the rodent head direction system". In: *Philosophical transactions of the Royal Society of London. Series B, Biological sciences* 369 (Feb. 2014), p. 20130283. DOI: 10.1098/rstb.2013.0283.

[PB18] Pereira, U. and Brunel, N. "Attractor Dynamics in Networks with Learning Rules Inferred from In Vivo Data". In: *Neuron* 99.1 (2018), 227–238.e4. ISSN: 0896-6273. DOI: https://doi.org/10.1016/j.neuron.2018.05.038. URL: https://www.sciencedirect.com/science/article/pii/S0896627318304367.

[Pou93] Poucet, B. "Spatial cognitive maps in animals: New Hypotheses on their structure and neural mechanisms". In: *Psychological review* 100 (May 1993), pp. 163–82. DOI: 10.1037/0033-295X.100.2.163. URL: https://www.researchgate.net/publication/14715295_Spatial_cognitive_maps_in_animals_New_Hypotheses_on_their_structure_and_neural_mechanisms.

[Sew20] Sewisy, A. E. *Biologically inspired spatial robotic mapping,localization and navigation using Spiking Neural Network*. Munich, Sept. 2020.

[Ska+94] Skaggs, W., Knierim, J., Kudrimoti, H., and McNaughton, B. "A Model of the Neural Basis of the Rat's Sense of Direction". In: *Advances in Neural Information Processing Systems*. Ed. by Tesauro, G., Touretzky, D., and Leen, T. Vol. 7. MIT Press, 1994. URL: https://proceedings.neurips.cc/paper/1994/file/024d7f84fff11dd7e8d9c510137a2381-Paper.pdf.

[Sol+09a] Solstad, T., Boccara, C., Kropff, E., Moser, M.-B., and Moser, E. "Representation of Geometric Borders in the Entorhinal Cortex". In: *Science (New York, N.Y.)* 322 (Jan. 2009), pp. 1865–8. DOI: 10.1126/science.1166466.

[Sol+09b] Solstad, T., Boccara, C., Kropff, E., Moser, M.-B., and Moser, E. "Representation of Geometric Borders in the Entorhinal Cortex". In: *Science (New York, N.Y.)* 322 (Jan. 2009), pp. 1865–8. DOI: 10.1126/science.1166466.

[Ste+12] Stensola, H., Stensola, T., Solstad, T., Frøland, K., Moser, M.-B., and Moser, E. "The entorhinal map is discretized". In: *Nature* 492 (Dec. 2012), pp. 72–78. DOI: 10.1038/nature11649.

[Tau09] Taube, J. "Head direction cells". In: *Scholarpedia* 4.12 (2009). revision #91349, p. 1787. DOI: 10.4249/scholarpedia.1787.

[Tau07] Taube, J. "The Head Direction Signal: Origins and Sensory-Motor Integration". In: *Annual review of neuroscience* 30 (Feb. 2007), pp. 181–207. DOI: 10.1146/annurev.neuro.29.051605.112854.

[Tau90] Taube JS Muller RU, R. J. "Head-direction cells recorded from the postsubiculum in freely moving rats. I. Description and quantitative analysis". In: *The Journal of neuroscience* 10.2 (1990), pp. 420–435. DOI: 10.1523/JNEUROSCI.10-02-00420. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6570151/pdf/jneuro_10_2_420.pdf.

[Tau84] Taube JS Muller RU, R. J. J. "Head direction cells recorded from the postsubiculum in freely moving rats. I. Description and quantitative analysis". In: *The Journal of neuroscience* 10 (1984), pp. 420–435. URL: https://doi.org/10.1523/JNEUROSCI.10-02-00420.1990.

[Tej+18] Tejera, G., Llofriu, M., Barrera, A., and Weitzenfeld, A. "Bio-Inspired Robotics: A Spatial Cognition Model integrating Place Cells, Grid Cells and Head Direction Cells". In: *Journal of Intelligent and Robotic Systems* 91 (July 2018), pp. 85–99. DOI: 10.1007/s10846-018-0852-2.

[Tol48] Tolman, E. C. “Cognitive maps in rats and men.” In: *Psychological review* 55 4 (1948), pp. 189–208. URL: http://psychclassics.yorku.ca/Tolman/Maps/maps.htm.

[Yod09] Yoder RM, T. J. “Head direction cell activity in mice: robust directional signal depends on intact otolith organs”. In: *The Journal of neuroscience* 29.4 (2009), pp. 1061–1076. DOI: https://doi.org/10.1523/JNEUROSCI.1679-08.2009. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2768409/pdf/zns1061.pdf.

[Zha96] Zhang, K. “Representation of spatial orientation by the intrinsic dynamics of the head-direction cell ensemble: a theory”. In: *Journal of Neuroscience*. 1996. URL: https://www.jneurosci.org/content/jneuro/16/6/2112.full.pdf.

[Zho21] Zhou, Z. “Biologically plausible spatial navigation”. MA thesis. Munich: Technical University of Munich, 2021.