André da Silva Pedro

THE HIPPOCAMPAL FORMATION FROM A MACHINE LEARNING PERSPECTIVE

UNIVERSIDADE DO ALGARVE

Instituto Superior de Engenharia

2020

André da Silva Pedro

THE HIPPOCAMPAL FORMATION FROM A MACHINE LEARNING PERSPECTIVE

Master in Electrical and Electronics Engineering
Information Technology and Telecommunications Specialization
Work performed under the guidance of:
Professor Doutor Jânio Miguel E. Ferreira Monteiro

UNIVERSIDADE DO ALGARVE
Instituto Superior de Engenharia

2020

The Hippocampal Formation from a machine learning perspective

Declaração de autoria de trabalho

Declaro ser o autor deste trabalho, que é original e inédito. Autores e trabalhos consultados estão devidamente citados no texto e constam da listagem de referências incluída.

____

André da Silva Pedro

Dedicado aos meus pais e à humanidade.

AGRADECIMENTO

Em primeiro lugar, gostaria de agradecer aos meus pais porque, por razões óbvias, sem eles nada disto seria possível.

Gostaria de agradecer ao Professor Jânio Monteiro pela sua disponibilidade, orientação e sugestões que me foram dirigidas ao longo desta Dissertação, bem como a todos os docentes que me acompanharam ao longo desta minha experiência de vida.

Um agradecimento a todos os meus amigos que nunca sentiram a minha ausência e a todos os meus colegas de mestrado, a destacar os meus colegas André Francisco e Joni dos Santos, que me ajudaram com algumas das suas ideias e momentos de descontração. Também gostaria de agradecer ao Nelson Pinto e Dario Cruz pelos seus conselhos durante as primeiras fases deste trabalho.

E claro, um agradecimento especial a todos os funcionários e membros da Universidade do Algarve, que me forneceu a formação e o acolhimento necessários durante os últimos 6 anos.

O meu grande obrigado a todos aqueles que, direta ou indiretamente, contribuíram para a realização desta Dissertação, bem como no auxílio à resolução dos diversos problemas da vida.

Faro, 24 de Setembro de 2020

RESUMO

Nos dias de hoje, existem diversos tipos de sensores que conseguem captar uma grande quantidade de dados em curtos espaços de tempo. Em muitas situações, as informações obtidas pelos diferentes sensores traduzem fenómenos específicos, através de dados obtidos em diferentes formatos. Nesses casos, torna-se difícil saber quais as relações entre os dados e/ou identificar se os diferentes dados traduzem uma certa condição. Neste contexto, torna-se relevante desenvolver sistemas que tenham capacidade de analisar grandes quantidades de dados num menor tempo possível, produzindo informação válida a partir da informação recolhida.

O cérebro dos animais é um órgão biológico capaz de fazer algo semelhante com a informação obtida pelos sentidos, que traduzem fenómenos específicos. Dentro do cérebro, existe um elemento chamado Hipocampo, que se encontra situado na área do lóbulo temporal. A sua função principal consiste em analisar os elementos previamente codificados pelo Entorhinal Cortex, dando origem à formação de novas memórias. Sendo o Hipocampo um órgão que foi sofrendo evoluções ao longo do tempos, é importante perceber qual é o seu funcionamento e, se possível, tentar encontrar modelos computacionais que traduzam o seu mecanismo.

Desde a remoção do Hipocampo num paciente que sofria de convulsões, ficou claro que, sem esse elemento, não seria possível memorizar lugares ou eventos ocorridos num determinado espaço de tempo. Essa funcionalidade é obtida através de um conjunto específico de células chamadas de Grid Cells, que estão situadas na área do Entorhinal Cortex, mas também das Place Cells, Head Direction Cells e Boundary Vector Cells.

Neste âmbito, o principal objetivo desta Dissertação consiste em descrever os principais mecanismos biológicos localizados no Hipocampo e definir modelos computacionais que consigam simular as funções mais críticas de ambos os Hipocampos e da área do Entorhinal Cortex.

PALAVRAS-CHAVE: Hipocampo, Aprendizagem Máquina, Células Grelha, Células de Local, Localização e Mapeamento Simultâneo

ABSTRACT

Nowadays, sensor devices are able to generate huge amounts of data in short periods of time. In many situations, that data, collected by many different sensors, translates a specific phenomenon, but is presented in very different types and formats. In these cases, it is hard to determine how these distinct types of data are related to each other or translate a certain condition. In this context, it would be of great importance to develop a system capable of analysing such data in the smallest amount time to produce valid information. The brain is a biological organ capable of such decisions. Inside the brain, there is an element called Hippocampus, that is situated in the Temporal Lobe area. Its main function is to analyse the sensorial data encoded by the Entorhinal Cortex to create new memories. Since the Hippocampus has evolved for thousands of years to perform these tasks, it is of high importance to try to understand its functioning and to model it, i.e. to define a set of computer algorithms that approximates it.

Since the removal of the Hippocampus from a patient suffering from seizures, the scientific community believes that the Hippocampus is crucial for memory formation and for spatial navigation. Without it, it wouldn’t be possible to memorize places and events that happened in a specific time or place. Such functionality is achieved with the help of set of cells called Grid Cells, present in the Entorhinal Cortex area, but also with Place Cells, Head Direction Cells and Boundary Vector Cells. The combined information analysed by those cells allows the unique identification of places or events.

The main objective of the work developed in this Thesis consists in describing the biological mechanisms present in the Hippocampus area and to define potential computer models that allow the simulation of all or the most critical functions of both the Hippocampus and the Entorhinal Cortex areas.

KEYWORDS: Hippocampus, Machine Learning, Grid Cells, Place Cells, Simultaneous Localization and Mapping

INDEX

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

AHA	*Artificial Hippocampal Algorithm*
AI	*Artificial Intelligence*
ANN	*Artificial Neural Network*
ART	*Adaptive Resonance Theory*
BVC	*Boundary Vector Cells*
CA	*Cornu Ammontis*
CLS	*Complementary Learning System*
DG	*Dentate Gyrus*
DQN	*Deep Q-Learning*
EC	*Entorhinal Cortex*
GPS	*Global Positing System*
HD	*Head Direction*
IOT	*Internet of Things*
LEC	*Lateral Entorhinal Cortex*
LSTM	*Long Short-Term Memory*
LTM	*Long-Term Memory*
LTP	*Long Term Potentiation*
MEC	*Medial Entorhinal Cortex*
MERLIN	*Memory, RL and Inference Network*
ML	*Machine Learning*
NEF	*Neural Engineering Framework*
QIS	*Quantization Intervals*
QTM	*Quaternary Triangular Mesh*
RESNET	*Residual Network*
RL	*Reinforcement Learning*
SARSA	*State-Action-Reward-State-Action*
SGD	*Stochastic Gradient Descent*
SLAM	*Simultaneous Localization and Mapping*
STM	*Short-Term Memory*

1. INTRODUCTION

1.1 INTRODUCTION

Human and non-human brains can associate many sensor inputs to predict the subsequent sequence of these inputs before they happen. In it, many inputs from the Visual, Olfactory, Auditory, Taste, Somatosensory, and Self-Motion senses are somehow merged, and combined, to form a coherent perception of our surroundings. The most interesting thing in this process is that this understanding is done without labelling these inputs, i.e. our brain can be seen as a black box that receives these signals and combines them to be able to identify the uniqueness of each event, both in a time basis and as a sequence of events, by a process that we call cognitive learning.

Nowadays, the ongoing developments in sensor and Internet of Things (IoT) networks are somehow reaching a level of complexity where the most complex of these systems are approximating the amount of data received by the simplest and smallest of brains that exists in living beings. The process of understanding how these brains work is thus very relevant to be able to build devices that can learn in a fully unsupervised manner.

Within the brain, the Hippocampus is one of the most interesting components. No one actually knew the capabilities of this organ until it was removed from a patient during a medical procedure [1]. Since then, this element has been the focus of much research in the biological context, but also in the computational context.

After the removal of this organ and studying the patient who participated in the clinical trial, it was clear that the Hippocampus had a significant role in the development of new memories [1]. One of the things detected in the patient was his lack of spatial and event notion, supporting the idea that the Hippocampus is an essential part in event recognition. Moreover, in the subsequent studies, it was shown that the Hippocampus plays a major role in information indexing and spatial navigation, by combining information from multiple sources of the individual's senses. The indexation of information allows the association of different events present in a specific memory episode.

In the Big Data's context, there are many situations where a lot of information comes from different sources but translate the same event. These so-called multimodal data have some similarities with the sensory information that enters the Hippocampus, and it would be relevant to create a model, or a set of models, that transform and analyse that information in a similar way as the Hippocampus does, i.e. that, without any type of previous training stages, is capable to adapt to the current conditions in the environment and has the ability to work with different types of data, with multiple dimensions.

1.2 DISSERTATION SCOPE

The scope of this dissertation is to find and report the major developments and scientific results in the scientific and computational area about the functioning of the Hippocampus, its internal components, and the associated interconnections. After the research of such biological and computational mechanisms, it is necessary to find how these elements work to produce a valid result. The objective of this work consists in building a computational system, or model, based in the understanding and research of the biological process involved and previous computational models, focusing on the navigational problem by associating different environmental elements to recognize and identify places in the environment.

1.3 DISSERTATION OBJECTIVES

Nowadays, due to the development of IoT networks, there is a lot of data that is being obtained and stored, making it imperative to create new methods that allow the extraction of information of these big datasets. In this context, the main objective of this thesis is to study the brain elements that are responsible for the creation of data representation and, encoding and if possible, to replicate their behaviour. A special emphasis will be given to Grid Cells [2] and the interaction between these cells and a virtual Hippocampus, trying to focus on a specific problem. Grid cells are a crucial element in animal navigation because of their ability to formulate grids based on the animals perception of the environment [2]. As the Hippocampus seems to be very much related with animal's navigation, we also focus on it. The implications of this work extend to many fields, including autonomous vehicles, as they require a navigational system like the one here developed, but also to machine learning and data analysis.

1.4 DISSERTATION STRUCTURE

The dissertation is divided into different sections that translate the analysis of the State of the Art, the work developed, the description of the obtained results and the conclusions. The State-of-the-Art section is divided into two different parts: (1) the Biological Part, containing information about the biological mechanisms involved in memory and spatial context creation and (2) the Computational Part that has information about strategies and algorithms used to replicate the biological behaviour.

The third section, "A Computational Model for the Entorhinal Cortex and Hippocampus", is also divided into two different parts: (1) the mechanisms implemented to simulate the Entorhinal Cortex behaviour, such as the creation of the Grid Cells, and (2) the Hippocampus part that has the strategy utilized to simulate its behaviour.

The information regarding the developed tests and the information is present in the Preformed Tests and Results section.

The Conclusions section analyses the results obtained and includes information related to the Future Work, together with the expected publications that will result from this dissertation.

The Appendix section contains special information about the conducted tests for specific modules.

2. State of the Art

In the last couple of years, a significant amount of research has been devoted to understanding the functioning and the interactions of the different elements in the brain. This process has been running in parallel with a significant amount of research that has also been devoted to Machine Learning (ML) and the algorithms used in this field of Artificial Intelligence (AI). While the first area of research tries to understand how our brain works, the second one tries to develop algorithms that artificially create what we will be calling cognitive intelligence. Approximating both areas is a process that deserves attention.

Inside the brain of humans and other mammals, there are two Hippocampi, one at each side.

Currently, a significant effort is focused on trying to understand this part of our brains. The main objective of the following sections is to report the major scientific work in the field of the Hippocampus and the associated Entorhinal Cortex.

2.1 A Brief Introduction of the Hippocampus

The Hippocampus is a brain element present in humans and other vertebrates, that can associate many of our sensor inputs to predict events place in a sequence [1]. The Hippocampus is an element that has gained a lot of attention because it is considered the responsible for the spatial memory and navigation of animals. The Hippocampus receives information that comes from many sensorial systems (Visual, Olfactory, Auditory, Taste, Somatosensory, Self-Motion) through the Entorhinal Cortex. Its outputs project back to the neocortex and to the Entorhinal Cortex. Somehow, these inputs can be combined to form an understanding of our surroundings by analysing and coordinating a set of features from the received inputs, i.e. our senses.

The Hippocampus has been associated with many functions, but most of them are related to memory and the process of self-motion. The first scientific achievements about the subject were made in 1957, when a famous patient, named in scientific community as H.M. (Henry Molaison), was stripped out of his Hippocampus, in an attempt to solve a problem related to seizures [1]. After the medical procedures, the patient's original problem decreased. However, his medical team concluded that the lack of the Hippocampus was directly associated with the subsequent inability to convert short-term memories to long-term memories, and with the loss of some long-term memories captured before the surgical procedure.

Several studies were made in different types of patients with different types of psychological diseases, like schizophrenia, various behaviour disorders, violence, and depression. As in the H.M. case, the patients' original problem slightly decreased or, in some cases, were completely removed, but they appeared to have the same side-effects as H.M. The medical team responsible by this study further concluded that the removal of the Hippocampus area prevents the conversion between short and long-term memories. Also, that the extent of the hippocampal removal can increase the level of memory loss, causing amnesia for the newest memories [1]. The personality and general knowledge of the individuals were still not affected [1].

2.2 FUNCTIONAL ANALYSIS

The Hippocampus, also known as "seahorse", is situated in the Temporal Lobe area. The main functions of this element are related to self-motion movements, emotional control, spatial navigation, and memory association.

One of the first models that tried to justify the hippocampal functioning was the memory index theory [3][4]. In the hippocampal memory index theory, the hippocampus is considered as being able to capture information about neocortical activity generated by the individual features of episodes. Using that information, this model assumes that the Hippocampus can then generate an index to the pattern of neocortical activity produced by an episode, projecting it back to these neocortical regions. Thus, the theory assumes that the features that make up an episode can establish a memory trace that activates patterns of neocortical activity. So, when a subset of a pattern is received by the neocortex, the projections from that input pattern can activate the connected neurons in the Hippocampus that represent the original experience.

The identification of that pattern by the Hippocampus is then projected back to the neocortex, activating the entire experience. Consequently, if a partial cue can activate the index, the associated neocortical patterns are activated and, with it, the memory of the episode is retrieved.

Besides the memory index theory, in recent years, several theories have emerged regarding the functioning of the hippocampus [5]. For instance:

(1) The Declarative Theory states that the Hippocampus works with the neocortical areas in order to access its data on a short-term period [6];

(2) The Multiple Trace Theory states that the Hippocampus works together with other brain parts to acquire and recollect episodic data. This task is exclusively done by the Hippocampus and the episode becomes strengthened by using the repetition method and the Long-Term Potentiation Method (LTP), allowing the memories to be accessed without the Hippocampus assistance [7];

(3) The Dual Process Theory considers that the Hippocampus collects data related to a specific episode and context by structuring them into smaller units. The recognition of the different units does not occur in the Hippocampus but happens in other Temporal Lobe areas [8];

(4) The Relational Theory, considers that the Hippocampus creates relations between events and acquired experiences that can be used in their identification [9]; and

(5) The Cognitive Map Theory, states that the Hippocampus builds representations of different aspects in the environment, creating a path between a specific goal and a starting point by using the obtained information from the different types of cells [10].

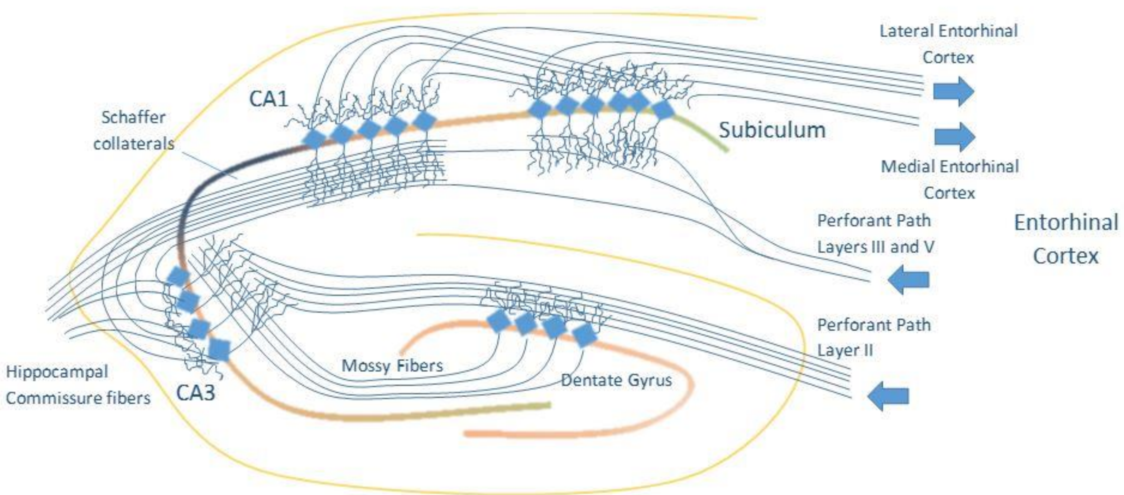


Figure 2.1: Biological Structure of the Hippocampus (adapted from [11])

2.2.1 STRUCTURE OF THE HIPPOCAMPUS

The Hippocampus structure is composed of different elements that can work at different levels. Figure 2.1 shows in detail how the different biological elements are combined, and Figure 2.2 illustrates a more simplified version of the interconnections between sections.

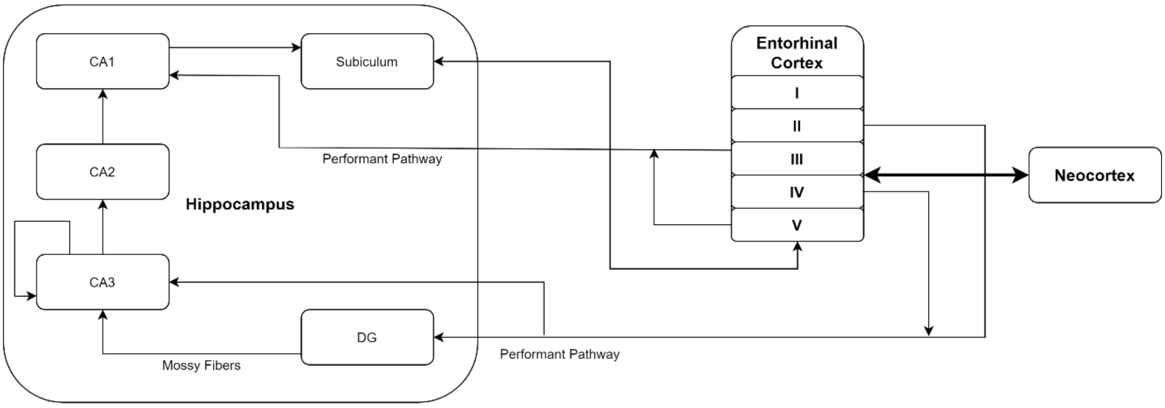


Figure 2.2: Structure of the Hippocampus and associated interconnections with the Entorhinal Cortex and Neocortex

The inputs of the Hippocampus are obtained from the sensorial system, via the Entorhinal Cortex (EC). Different layers of neural fibres (II, III, IV and V) project from the EC to the Hippocampus. Each of these layers enter the Hippocampus in different places, through the Perforant Pathway.

One of the main elements of the Hippocampus is the Dentate Gyrus (DG). The DG structure is considered to work as a pattern separator element, meaning that this element classifies the data provided by the EC [12]. Some studies have shown that the DG allows the discrimination of the location of two identical objects, with a certain physical distance between them [4].

Another set of important components are the *Cornu Ammontis* (CA) elements. The CA region is a strip of pyramidal neurons that is divided in different areas, numbered from 1 to 4 (i.e. CA1, CA2, CA3 and CA4). CA areas are considered responsible for the association between elements, creating episodic sequences and using those episodes to acquire new information [6].

Pyramidal neurons, as other neurons, are composed by dendrites and axons. They can be divided into different parts, where each part receives a unique input. The cells also have thin spines and it is believed that those spines are responsible for the learning process, by connecting different types of neurons using the Hebbian method (see section 2.2.2 for more information), while the larger areas are responsible for memory storage and management [7]

[14]. The receptors of these neurons are complex because the inputs are sensitive to changes in the environment. The behaviour of pyramidal neurons makes them suitable for the processing of new information [14].

It is believed that CA elements are responsible for the prediction of the next steps [15]. The connection between the Layer III of the EC is necessary to make such predictions when the individual is extracting environmental information [15].

The CA3 subarea is the first of the CAs and is considered as one of the most important CAs. In terms of connections, CA3 receives connections from the DG via a group of fibers, called mossy fibers, and from the EC, via the Performant Pathways. The CA3 is considered as responsible for the retrieval of the information and its transformation into short-term memories [12]. It also seems to complete the pattern separation from the previous element [12]. The transformation is made by associating a specific object or task into a specific element. The auto associative connections in the CA3 area are illustrated in Figure 2.1, and Figure 2.2 allows the recall of specific events, creating a memory of the event [16].

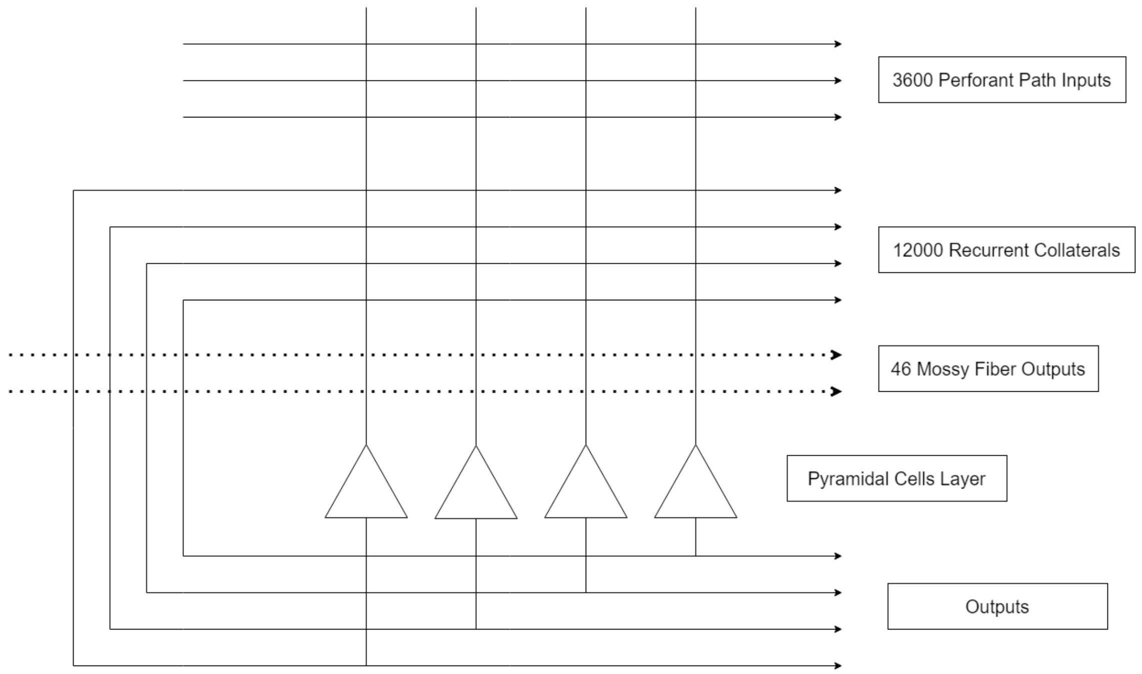


Figure 2.3: Schematic of Dimensionality of the CA3 in the rat, adapted from [17]

Figure 2.3 shows the connections of a CA3 element present in the Hippocampus of the rat. The CA3 element has many recurrent connections. In [17], it is considered that this may occur because, in some cases, it is necessary to add more features to complete information about a specific context. The decisions about the retrieval of the information are made inside the pyramidal cells. The inputs sent to the CA3 are sparse and they contain information created by the separation of the patterns and the information sent via EC [17].

The following CAs are the CA2 and CA1. The CA2 subarea has multiple pyramidal neurons that receive inputs from the CA3 and DG areas, and it is responsible for passing information between the CA3 and the subsequent CA1 subarea. According to the "*Rediscovering area CA2: unique properties and functions*" [18], the pyramidal neurons in CA2 carry less information than other pyramidal neurons.

Regarding the CA1 area, it seems to be associated with the retrieval of specific events in the individual’s memory, which means that the CA1 neurons can recognize and reflect the stored memories.

According to [13], both CA3 and CA1 are involved in memory for sequential non-spatial events that compose unique experiences, and these areas play different roles that are distinguished by the duration of time that must be bridged between key events. Specifically, the findings suggest that the role of CA3 in sequence memory is not limited to spatial information, but rather appears to be a fundamental property of the CA3 function. In contrast, CA1 becomes involved when memories for events must be held or sequenced over long intervals.

The CA3 and the CA1 are deeply connected by a set of neurons called Schaffer Collaterals and it is believed that these neurons allow the creation of a topologic representation between the CA areas [19]. The lamellar hypothesis states that the cells from the EC area organize the information sent to the CA’s using different types of lamellas and each lamella has specific information regarding one type of cell. This organizational structure is present in the CA’s element and in the DG area [20].

After the CA areas, there is an element called Subiculum. The subiculum is composed of 3 different subareas: (1) An area that connects directly with outputs of the CA1, (2) a Pyramidal Neurons area and (3) the area that connects with the EC. Research that study lesions in the subiculum suggests that the Subiculum plays an important role in spatial memory [21]. It can assist the Hippocampus, in order to form a long-term memory and it can also enhance the Hippocampus output [21]. According to [15], the Subiculum can also be an input of the Hippocampus sending information regarding the boundaries of the environment and that information can be used during the phase precession stage, described in section 2.2.4.

As shown in Figure 2.1, there is a set of connections called the Hippocampal Commissure Fibers. According to the “*Uncovering a Role for the Dorsal Hippocampal Commissure in Recognition Memory*”[22] , such fibers connect both Hippocampi. The information regarding the other Hippocampus enters directly into the CA1 area. According to "*The origin of the hippocampal commissure in the rat*" [23], the CA1 area is the beginning of the Hippocampal Commissure Fibers, suggesting that CA1 areas of the two CA1s, in both Hippocampi, are connected.

2.2.2 HEBBIAN THEORY

As previously described, it is considered that Pyramidal neurons use the Hebbian method in the learning process [14]. The Hebbian method seems to be involved in the synaptic integration of these Pyramidal neurons [14]. Integration is the mechanism by which a subset of inputs can trigger a pyramidal neuron to fire.

The Hebbian theory marks the start of the Artificial Intelligence connectionist approach, claimed by Donald Hebb. He presented a theory on the functioning of brain's synapses based on investigations made by Charles Sherrington and Ivan Pavlov. After their observations, Hebb concluded that "When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process, or metabolic change, takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased" [24]. The theory is often summarized as "Cells that fire together, wire together." [25]. It is thus considered that an excited neuron decreases the electrical discharge to neurons that are not active at the same time, while increasing the discharge to neurons that are active at the same time. With the electrical discharges, it is possible to create a path of active neurons and the repetition of such firings strengthens and facilitates the creation of a neural route. After a few years of study, Hebb created his postulate: "The synapses increase their strength only if there is a post-synaptic neuron firing by other type of input causing an electrical impulse in the neuron". Based on his theory, he created computer models that would refer to this behaviour as a Hebbian Synapse. These synapses can be created by using logical ports such as AND gates with a set of two types of inputs, and with those inputs a system can associate different memories from different contexts [24].

2.2.3 INPUTS AND OUTPUTS OF THE HIPPOCAMPUS

The Hippocampus receives data from different types of sources, and that information is crucial to interpret the current state and to make the necessary predictions. The main input and output of the Hippocampus comes from the EC.

The result of the Hippocampus calculations is modulated in a specific wave type. That wave is called Theta wave. It has a specific frequency of transmission and, according with [26], the job of the EC is to interpret the information in it. The EC thus acts as an intermediary element between the Hippocampus and the different areas in the brain. It may act as a reinforcer of the hippocampal outputs [27]. The EC is divided in two separate areas, the Lateral Entorhinal Cortex (LEC) and the Medial Entorhinal Cortex (MEC). As stated in [28], the LEC area seems to be responsible for combining spatial and non-spatial information, while the MEC area creates a set of landmarks essential to path integration.

The LEC seems to encode time information in different time scales across different episodical contexts. As stated in [29], the information created on the LEC is sent to the Hippocampus, giving the “what”, “when” and “where” representation of a specific event. On those areas, there are several layers of cells directly connected to each other.

The MEC area has a set of different types of cells that are responsible for the path integration mechanism. Among them, are the Grid Cells [2] and the Speed Cells [30]. As the name suggests, the Speed Cells encode the subject’s velocity information by firing their cells in a linear rate.

A major part of the Hippocampus inputs come directly from the second and third layers (Layers II and III) of the MEC. Layer II connects to the DG area. Layer III projects directly to the CA3 and associated areas [27].

Layer II contains the largest amount of Grid Cells, that seem to provide and create an internal map of a specific local environment. For each local environment, there are specific firing procedures. The combination of several Grid Cells allows the animal to identify a specific point in space [2]. According to [2], Grid cells can be seen as a path integrator of a certain space. Also, the firing pattern of Grid Cells is displayed an hexagonal form, when put into a 2D environment [2]. Yet, the association between the real-time data and the places already known by the animal, happens in the Hippocampus.

2.2.4 HIPPOCAMPUS CELLS

Inside the Hippocampus, there is a specific group of cells that is responsible for the association of several features, such as location, direction, the boundaries of the space where the animal is, and also the current time. These are, respectively, the Place Cells, Head Direction Cells, Border Cells and Time Cells.

Place Cells are a set of hippocampal cells that fire when an animal visits a location in an environment, allowing the creation of a cognitive map of the different locations. According to [31], Place Cells have a direct relationship between the CA1 identified location and the location of a certain goal. Such behaviour is achieved by an increase of the firing rates when the animal is moving towards a possible goal. Those firing rates are determined by the routes or future routes, which can lead to a specific goal [31]. These cells are present in many different brain elements, especially in the CA1 area, and can rapidly adapt their place fields when there is a change in the environment. The place field consists in the spatial field that is represented by the place cell. Different Place Cells encode different parts of the environment, but the sequence of places is not directly represented inside the Hippocampus. The replay of the steps is made when the individual reaches its destination or during sleep. The recall of the steps is crucial to form long-term memories. Inside the Place Cells, there is a phenomenon called Phase Precession. As mentioned before, the Hippocampus produces a wave called Theta wave and each Place Cell has its own area that is called a place field [32]. When the subject is in a specific area, that area is recognized and produces a set of spikes during the phase of the theta signal.

According to *"How Animals find their way in Space. Experiments and Modelling"*[32], if the subject stands between two place fields, both Place Cells will spike, but when it reaches the centre of one area only the cell that represents such area will spike. This mechanism allows the recall of the given steps, in which case strengths memory connection necessary to encode spatial information. Figure 2.4 illustrates the mechanism of phase precession with an animal crossing a one-dimensional environment.

According to Figure 2.4, the Phase Precession occurs when an individual moves between place fields and according to [15], it is not clear how the phase mechanism works. This system allows the compression of spatial information by using timing dependent information extracted from the generated waves [15].

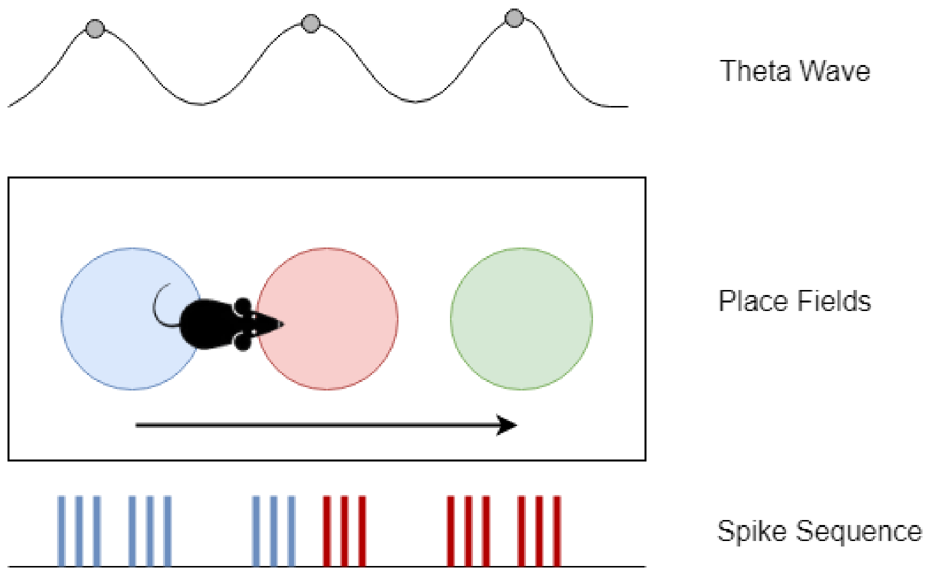


Figure 2.4: Phase Precession Behaviour adapted from [15] and [32]

The Head Direction (HD) Cells are a group of specific cells that are responsible for translating the direction of an animal on the environment. These cells, with the help of other elements in brain, work as a brain compass and are activated when the head of the individual is facing a specific direction. The HD cells collect data from different senses to update its direction, maintaining the spatial map created by grid cells and place cells [33]. In "*A dual axis rotation rule for updating the head direction cell reference frame during movement in three dimensions*"[34], it is suggested that HD cells rotate when the subject is facing an obstacle like a wall. When this behaviour happens, one of the axes has an opposite direction of the other, proving that HD cells have a dual-axis system that allows the representation of complex planes while maintaining a stable notion of direction. These types of cells are present in the EC, Subiculum, and in another brain element external of the Hippocampus called Thalamus. "*Encoding of 3D head direction information in the human brain*"[33] indicates that the Thalamus has a major role on the encoding of the horizontal heading on 2D and 3D environments. The Subiculum also plays its part on direction finding. According to [33], there are HD cells with the capability to explore a volumetric space and to make azimuth calculations. HD cells use referential landmarks, particularly those captured via visual stimulus, to choose their preferred location. The HD cells can: (1) choose their preferred location based in bi-directional compartments, (2) choose the opposite location of the bi-directional compartments and (3) fire in a bi-directional way [32].

The Border Cells, also called Boundary Cells or Boundary Vector Cells (BVC), are a group of cells that fire when the animal or human finds a boundary on the environment. These cells, located on the Subiculum, allow the extraction of information of edges and borders giving the notion of a closed space. They work autonomously from the HD cells. On the other hand, border cells need to work with place cells, allowing the recognition and extraction of features from the current environment. The distance is the main feature of these cells, because border cells fire when the subject is near the wall or close to another type of obstacle that may be present in an allocentric direction [35]. These cells can be found in the EC and the information encoded is sent to the Hippocampus for further analysis [35].

Time Cells are specific cells that fire on specific periods of time [16]. The sequences created allow the encoding of temporal information with precision. The macro time information can be retrieved by drifting the Time Cells. Those drifts synthesize different aspects of temporal coding by indicating a set of neurons that reflect time scale information. Individual "time cells" can be active or inactive in different sequences, which represent different time periods. According to "*The Same Hippocampal CA1 Population Simultaneously Cods Temporal Information over Multiple Timescales*"[36], the experiment made indicates that Time Cells sequences rarely change daily and that they produce temporal signals over different timescales. During their research, the authors found that cells responsible for retaining temporal information stabilize over two consecutive days [36]. The Hippocampus requires the information provided by the Time Cells to adjust the data into a sequential set of events. It is also believed that Place Cells can have the same role as Time Cells, because those cells fire when the subject passes through different areas of the environment at different times [32].

2.2.5 Biological Models of the Hippocampus

After the discovery of the importance of the Hippocampus on the brain system, several models were developed with the aim of creating an artificial Hippocampus and to further understand its functions and capabilities.

The Marr's model [37], named after its creator David Marr, states that the Hippocampus is a primitive element from the archicortex. The archicortex is a simpler part of the neocortex, because it has fewer cell layers and may have a small number of connections. In this model, the neocortex learns information provided from sensorial data and the archicortex does the association, memory indexing and recall. The Hippocampus is seen as an element that can train the neocortex and transform short-term memories to long-term memories. Marr believed that a system that could simulate the hippocampal memory system would need to store 105 events and he also suggested that Hippocampus was responsible for the storing of simple representations of the events. This model uses sparse codification. In it, it is possible to create inhibitory mechanisms and enable the prediction of data by using the retrieval method. The model does not follow the Hippocampus information circuit and it is not clear how the Hippocampus transfers memory [37].

O'Reilly & McClelland [38] created a model that uses the Hippocampus elements and information flow. According to their findings, the learning process happens between the EC, DG, CA3 and CA1. The second layer of the EC sends data to the CA3, creating a new representation that facilitates information storage and retrieval. The third layer of the EC sends data to the CA1 to form an invertible representation that could recreate a specific EC pattern. On their models, an “winner takes all” approach has been used to select the best cells that represent a specific event or episode. The pattern overlap is calculated by computing the number of cells in common. If this number surpasses a certain threshold, a pattern separation is required [38].

2.3 COMPUTATIONAL ANALYSIS

In recent years, technology has evolved in a way that old theories can be put to test, while new models for biological systems are developed. Computational Scientists and Biological Scientists are combining resources to create systems which can replicate biological behaviours. AI can be seen as a set of calculations used to achieve specific goals or we could say that AI is a way to solve real world problems by creating machines with biopsychological characteristics [39]. Associated with AI, there is a field area called Machine Learning, which can be viewed as the system ability to gather, interpret, and extract information using real world data [40].

Throughout time, many ML and AI algorithms were developed to solve the world problems by learning with the acquired data. The ML learning algorithms can be described in three different categories: (1) Supervised Learning, as it is a method that uses associations and Statistical Properties in order to distinguish objects or elements, (2) Unsupervised Learning, where the machine must learn and acquire the different features from a dataset and (3)

Reinforcement Learning (RL), which can be seen as a method that produces a set of decisions, with a score, which can affect the output [40]. In this field of AI, the learning process is directly extracted from the interactions within the environment, by analysing its actions with rewards or punishments.

ML algorithms require an understanding of the available data. Not all algorithms work on any specific dataset and that is the main reason why there are multiple ML algorithms for different types of problems. To make use of a ML strategy, it is necessary to ensure that the dataset has enough information to answer the questions or to allow the prediction of possible outcomes [41].

There are many frameworks and libraries that can be used to simulate such behaviours and one of them is the Neuron Engineering Framework (NEF)[42] . The NEF is a framework built in large scale that can model the neurons behaviour to simulate the cognitive mechanisms by using a set of different feedforward neural networks. The NEF is suitable for online, unsupervised, and supervised learning [42]. The NEF represents its data by mapping the time variant data into the neurons. The decoding method is implemented with the help of the least-squares method [43]. According to [42], transformation from time data into neural data, allows the framework to use neurons and directly work with them, to obtain the results by computing a non-linear function using the weight matrix of two neural populations. With the NEF, it is possible to build various types of models purely based on biological systems. The Nengo Python library implements the NEF characteristics and can work as a neural compiler that sends its commands directly to the NEF system [42]. With Nengo, it is possible to simulate different behaviours without the usage of big server farms and computational resources [42].

As mentioned in the biological analysis, one of the Hippocampus' functions is associated with spatial navigation. There is a specific group of algorithms that allows the simultaneous localization and mapping (SLAM) within an environment with the help of landmarks. While there are many types of SLAM algorithms, they all solve the same problem, i.e. the ability to map a large or smaller environment that could be indoor or outdoor. SLAM systems are crucial for mobile robots and autonomous vehicles [44].

Nowadays, the AI approach offers several solutions various problems. The Reinforcement Learning and other types of AI approaches could be good solutions for SLAM algorithms, because it is possible to map an environment using a trial-and-error method or other types of mechanisms. However, the approaches in AI are purely computational or genetic and do not regard any neurobiological aspects. That is the main reason why different types of frameworks are developed. With Nengo framework, it could be possible to solve the SLAM or other types of problems without using any complex mathematical equations.

2.3.1 Computation Models of the Hippocampus

Because of the Hippocampus role on the learning system, there are several computational models that simulate the biological behaviour of this brain element.

Among the models that are based on the Marr's Model, the Complementary Learning System (CLS) model characterizes the learning organization of the brain [45]. According to it, the Hippocampus and the neocortex are in constant communication. The Hippocampus is responsible for learning new information and it is also capable of replaying the previous information by taking advantage of the pattern separation made in the DG element. The central support of CLS model is that the Hippocampus encodes information differently from the EC [46]. Also, the CLS introduced the "big-loop" question. The "big-loop" question happens when the pattern reconstructed by the EC is sent back to the Hippocampus region as input. The Hippocampus must make some statistical calculations to find patterns across episodes [47].

The BBB Model [5], created by Byrne, Becker and Burgess, is a computer model that consists of a set of connections and neural mechanisms that are present in the medial temporal, parental and prefrontal lobes. From the model's point of view, the Hippocampus plays a major role on representing spatial information by collecting the data, examining the visual features and head direction data coming from the HD cells. Also, the Hippocampus creates a spatially coherent image of a remembered scene. However, this model does not contemplate the processing of non-spatial information.

The Memory, Reinforcement Learning, and Inference Network (MERLIN) [48] consists of a method that benefits from the Reinforcement Learning (RL) technique, and makes use of a specific type of Artificial Neural Networks (ANN) called Long Short-Term Memory (LSTM) working as an external memory. The creation of this model was based on the knowledge of the Neuroscience and the Psychology. The information is encoded based on the temporal context model by using Residue Network (ResNet) modules with different types of layers. This model is good to solve navigational problems, it can localize goals on unknown environments and this achievement can be accomplished by a hierarchical goal system [48].

The Artificial Hippocampal Algorithm (AHA) [47] is an algorithm that tries to create a digital Hippocampus. This model uses the EC, DG, CA3 and CA1 elements from the Hippocampus and tries to emulate its functions using AI techniques like ANN. To test this method, a One-Shot Learning dataset was used. The EC creates a sparse distribution that sets together with the different features from the input. The DG has the function to sparse the EC input and to separate the data to find simpler patterns, so the DG element can work as a hash function because similar inputs must create dissimilar outputs. It produces an orthogonal representation that can be seen on the EC-DG-CA3 pathway. In this model, the CA3 connects directly to EC, allowing the creation of an association network, and this network is responsible for recalling data from the CA3. The CA3 element is implemented with the Hopfield network [49] [50] because it is biologically inspired, auto associative and it can address memorable content. CA3 stores the data from the DG and must recall the right subset of the EC data. The EC-CA3 connection allows the exploitation of the data represented in EC in which it is overlapped and not separated [47].

In "*A Brain-Inspired Goal-Orientated Robot Navigation System*" [51], a different model of the Hippocampus was used. The Adaptive Resonance Theory (ART) is a neural theory that explains the brain's capability to classify and recognize different events on changing environments. The ART model is a biological-plausible theory that uses ANN with feedback as a primary mechanism to calculate the results. In this model, the problem between stability and plasticity is present. Plasticity is the ability to adapt from different situations, but the adaptation can lead to instability on learning the different features of the environment, causing a forgetting of the information retained. Stability means that collected states rarely changes and it is possible to create different types of clusters. In the ART methodology, the capacity to learn different types of arbitrary patterns will attenuate the issues referenced before. This method can be used in different types of supervised and unsupervised ML problems. The ART model can work as a memory model because, with the connections of different neural network layers, it is possible to exchange data between short-term memory (STM) and long-term memory (LTM). In ART, a competition method is used, meaning that neurons with more activations are the ones that are used to analyse data and get the results. There are different levels of ART implementations: ART1 creates a cluster by using binary input samples and calculating the similarity measure using a system based on the Hamming distance. The ART1 allows online learning, capture of rate events, evaluation of subset and superset patterns and biasing the neural network to create new episodes and contexts. The ART2 model is an extension of the ART1 by allowing the usage of real data. The ART2 was substituted by Fuzzy

ART, which uses Fuzzy Logic in conjunction with ART neural networks. After the Fuzzy ART, the ART3 and ART2-A were developed. The ART2-A is a faster version of the ART2, and the major characteristic is the adaptation the weights used during the data computation [51].

The RatSLAM system uses the hippocampal system to represent and create a map of a certain environment. This algorithm uses the concept of place fields, fields responsible to identify specific areas on the environment. The basis of RatSLAM system is the usage of competitive attractor networks. The movements of the subject, in this case a robot, are represented on those networks. The movement of the robot in conjunction with the sensorial cues allows the network update and consequently the encoding of a new position. With this system, it’s possible to extract untreated or noisy data because it is the responsibility of the network attractor to select the best hypotheses by using the competition. The competition system will give more credit to the most probable outcome [52]. The RatSLAM system is composed of different parts:

(1) The Local View module, responsible for the calculation of distance and the relative bearing to a place by using the extracted visual data. The local view module is composed of different cells, called local view cells, and those cells are placed in a three-dimension matrix.

(2) The Pose Cells module uses the competitive attractor neural network. The function of the neural network is to create and converge to a stable representation of the cell activation. The cells of the network can be excited or inhibited and use the neighbour excitation method, cells near the cell represented are excited and the further ones are inhibited. The position and head direction information are calculated and encoded, using separated networks. The arrangement of the network is highly flexible and can facilitate the Path Integration.

(3) The Path Integration correlates the various bearings and distances made by the individual. The cells are placed in the environment in a specific way and directly depend on the Pose Cells result.

RatSLAM can encode different areas without the cartesian representation. Instead, it uses a topological one. However, as time passes, this model can suffer from drift originated from the noisy data obtained by the pose but on other hand this method is not sensitive to ambiguous information. One of the problems related to this algorithm is its usage on larger environments, because it requires memory capacity to maintain the Pose Cells representation [52].

The Whisker – RatSLAM [53] is an extension of the RatSLAM system. This method uses a set of sensors called whisker sensors. Those sensors scan the floor of the environment and use that information to establish the landmarks and include them into the experience map. The sensors will update its values as the robot moves and when they find an obstacle, the algorithm will enter an exploration mode. On the exploration mode, a 6-dimensional map is created, and the system will calculate the best route to avoid the obstacle. In conjunction of the Pose Cells, a new set of cells are used. The Feature Cells encode information relative to the observed object by using statistical information regarding the region observed by the whisker sensors and the different slopes associated to the observed object.

2.3.2 COMPUTATIONAL REPRESENTATIONS OF GRID CELLS

As mentioned in section 2.2.3, Grid Cells are a group of cells placed in the EC element, with the responsibility to integrate the individual's path. The position of Grid Cells suggests the formation of a hexagonal pattern. In computational, systems the hexagonal grids are used in many applications, one of them being in spatial representations of games environments. The hexagonal grid applied to computer games has the advantages of: (1) maintaining a consistent connectivity between distinct hexagons, (2) the distance between neighbours is the same and (3) the edge detection becomes easier. The hexagonal coordinate system is similar to the cartesian system. The X axis represents the horizontal direction, the Y axis represents the vertical direction, and the Z is the negative difference between X and Y axis. By using this system, it is possible to use plane operations like rotations, scaling and reflexions [54].

According to "*Computational Models of Grid Cells*" [55], there are different types of grid cell models, namely: (1) the Oscillatory-Interference Model; (2) the Attractor-Network Models; (3) a Self-Organized Model and (4) the Path-Integration Model.

The Oscillatory-Interference Model is based on the idea of the phase precession method. Two oscillators are implemented, and one has the function to maintain the frequency constant while other increases or decreases frequencies based on the input. When a threshold is applied, it will result in a phase change, causing a difference between the oscillator and velocity. One side effect of this model is the creation of repeated place fields that were not seen at the time. When the difference between oscillators is a multiple of 60 degrees, a triangular grid is created. However, this model is not plausible on a biological system because the implemented oscillators have perfect sinusoids and, with noise, the performance of the model is significantly lower. One method to solve this issue is updating the network with environmental landmarks. The firing pattern of the grid cells can be maintained for a period of ten minutes before complete darkness. The stability of the model is obtained by connecting the oscillators using inhibitory and excitatory entries.

The Attractor-Network Models is another computational representation of the Grid Cells. A single position in the grid is represented as an attractor with an activated state and using recurrent connections. A specific set of inputs might activate the neurons in the net. If the recurrent connection is stronger than the connection of the neighbours, the spike will occur in a different part of the net and the incoming information directly influences the location of the spike. The information encoded via HD cells can be implemented using an element called a one-ring attractor, by putting the different cells in a circular form and according to the preferred direction. This model can also be used to represent the behaviour of the entorhinal network. This navigation model, used over different periods of time, allows the creation of Grid Cell patterns, but it requires a boundary to act as an identifier of environmental boundaries and an activation function responsible for the control of firing of the initial connections, in order to avoid a possible overfitting of data. To add the temporal information necessary to create a phase precession, a tours-based model has been proposed. Two connected networks are used to simulate the theta rhythm wave and with this feature it is possible to follow the movement and direction of the individual. This model was made based on the assumption that the phase of the wave that has spatial information is directly linked to a spike on the Grid Cell network.

A Self-Organized Model consists in creating an emphasis on inputs that are separated by a 60-degree angle. In this model, it is believed that grid cells receive data from a group of cells called stripe cells. The stripe cells fire different stripes into the environment and the junction of the linear velocity, and the allocentric direction allows the creation of constantly spaced paralleled lines. The direction vector is perpendicular to the stripe cells lines [56]. The ring attractor network is used to extract data from the velocity signal and the movement of the individual is reflected into the stripe cells by using a competitive learning process. With this method, it is possible to generate the hexagonal pattern. The Hebbian learning method can be used to implement the competitive learning by giving strength to cells that have a 60-degree orientation. This model has features from the models mentioned above and its main idea is that cells can create stable hexagonal patterns on a short time.

The Path-Integration-Free Model suggests that the formation of grid cells is only made with spatial and velocity inputs. This model also uses Hebbian learning to form/ organize the network and the neurons must have a special feature to allow the adaptation of different situations like the average of the velocity multiplied by a time factor. In this model, the grid fields might have random orientation, and these can be placed by using excitatory recurrent connections.

These models make a number of hypotheses, that can be tested in the future. Until now, the results obtained from experimental evidence are subject to multiple interpretations. As improvements are made in such methods, the required alignment between theory and reality will gradually be strengthened. While these improvements in experimental methods do not allow verification, it is important to define computational models to provide a fundament for the subsequent testing, as these methods become available.

2.3.3 REINFORCEMENT LEARNING

As mentioned in section 2.3, the Reinforcement Learning is an emergent field area of AI. According to the book "*Reinforcement Learning: An Introduction*" [57], the Reinforcement Learning (RL) can be seen as a field of the Artificial Intelligence who tries to map situations into actions, in order to maximize a certain outcome. These types of algorithms are currently used when there is no information about the environment. From that, they conclude that RL Algorithms are mostly used when exploration and exploitation are needed to achieve a specific set of goals. RL is the closest type of learning that animals and humans do on their entire lives [57]. According to "*More than the Sum of its Parts: A Role for the Hippocampus in Configural Reinforcement Learning*" [58], there is a direct connection between the Reinforcement Learning area and the Hippocampus. The Hippocampus biological location allows the creation and classification of memory events. A RL method can use this kind of match making strategies to reinforcement and distinguish good actions from bad ones. Together with other brain elements, the Hippocampus is an essential part of a bigger biological RL system that uses our emotions to balance behaviours in a certain environment and the RL system can be the information analyser for the decision-making process [58].

According to [59], there are two types of RL models: Model-based RL and Model-free RL. Model-based RL is a model that uses experience to learn, do the transitions and the actions on a certain environment. This type of model works like a mental map and searches the actions of the agent, based in the actual state and possible outcomes towards a specific goal [59].

Model-free RL is a model to encourage the trial-and-error method. It is easier to implement. However, it requires a lot of experience to achieve the goals, because this type of model uses the state to predict outcomes and does not use the environmental variables [59].

2.3.3.1 REINFORCEMENT LEARNING ALGORITHMS

On the Reinforcement Learning field, there are multiple algorithms that fit on the approaches mentioned before. The most common algorithms are: Q-Learning, Deep Q-Learning (DQN) and State-Action-Reward-State-Action (SARSA).

The Q-Learning algorithm is a model-free RL algorithm that estimates the next action based on the current state and possible outcomes [60]. This algorithm reaches its goals by making the expected sum of the future rewards. For each possible state, this method calculates its rewards and then finally selects the maximum value. Associations between states and outcomes can be represented in a table format. The Q-Learning algorithm has several steps: (1) acquires the current state, (2) chooses an action to execute, (3) observes the consequent state and (4) adjusts the rewarding table values and updates the values from the Q-table. For each iteration of the Q-Learning algorithm, new associations will appear due to the constant revaluation of the actions and consequently the updates to the Q-values [61].

The Stochastic Gradient Descent (SGD) method can be used to update the combination of the state and the action values. The actions represent the necessary movements to reach the states. On the other hand, this algorithm suffers from overestimations and those overestimations occur if the action values have a uniform distributed error. The Double Q-Learning algorithm was created to solve this problem. In this method, the operator used for maximizing the outcome uses the same values to evaluate the next action and also uses two sets of weights: one defines the exploration and exploitation factor (greedy search) and the other is responsible for the determination of its value [60].

The DQN, which is a combination between Q-Learning Algorithm and Deep Neural Networks, uses neural network to estimate some parameters, called weights, that are used to make the decision and maximize the reward. The Deep Q-Learning also needs some type of memory. The addition of a memory allows the algorithm to learn from previous experiences. This experience replay improves the efficiency, enabling obtaining better results [60].

The SARSA algorithm uses the Bellman Equation to estimate the action to estimate all possible actions in an environment [62]. The update values are constructed at the next sample time. The exploration and exploitation factor, on this algorithm, only depends on the Q value. In the most common cases, SARSA will not converge to an optimal or near optimal value, except when the greedy factor varies inversely with time [62].

2.4 FINAL CONSIDERATIONS

The analysis of the different types of biological and computational models of the Grid Cells, present in the Entorhinal Cortex, lead to some considerations. Regarding some biological subjects or elements, there is no consensus. Some authors create models according to their assumptions, while others only create assumptions without any type of proof or valid test. The uncertainly of the possible behaviours makes the creation of a possible computer model difficult because scientists rely on the existing studies of possible behaviours, which do not necessarily translate the complete set of associated features. However, despite the many uncertainties, there are some things that biological and computational models have in common. All the Grid Cell models use velocity and orientation in the process of creation of the cells.

In the Hippocampus models, there is more consensus. Most of the Hippocampus biological and computational models must have elements that allow the division of information into several parts, perform pattern separation, and some kind of system that analyses the information and recognizes some characteristics of the location (place) or events already processed. The Hippocampus elements like the DG, CA1 and CA3 areas, are present in most of the analysed models. Some of them implement different types of cells, such as the HD cells or even speed cells. The main difference between the modules is how the elements connect and share information and how they work to produce a valid result.

3. A COMPUTATIONAL MODEL FOR THE ENTORHINAL CORTEX AND HIPPOCAMPUS

After the analysis made in the previous sections, this chapter focuses on defining and implementing a model for both the Entorhinal Cortex and the Hippocampus. The chapter is divided into two parts. The first part contains the model created for the definition of Grid Cells, allowing the encoding of a specific location in the environment. The second part focuses on the functioning of the Hippocampus and how the different inputs can be combined there to return a valid result. The purpose of this model is to explore how the Hippocampus and the Entorhinal Cortex could recognize and divide an environment into different places.

3.1 MODEL GUIDELINES

As mentioned before, in most of the research that involve the Entorhinal Cortex and Hippocampus there is not a consensus about different aspects, such as the functioning of its elements and their connections. In the models proposed for these elements, we also need to rely on some working principles and requirements.

In the model here proposed for the EC, we assume that the Grid Cells receive the speed and head direction as inputs. The provided elements from the EC are not distinguished from LEC and MEC areas. As for the Hippocampus, it will be composed by the following elements: DG, CA3, CA1 and Subiculum. The proposed model shares a similar structure as the Artificial Hippocampal Algorithm, as mentioned in section 2.3.1. However, the model does not implement any type of ANN. Unlike the analysed biological models there is not a connection between the Subiculum and the Entorhinal Cortex. In section 2.3.1 the Subiculum element was not taken into consideration. In this model, it is assumed that the Subiculum will create some type of mapping that connects the different Place Cells. This assumption is supported by the fact that the Subiculum has different types of cells, including the BVC the HD cells [63]. Those cells could help in the formation of a self-awareness map [63]. The creation of a map could be an essential piece of information for the brain's decision-making elements. However, to create the map it is necessary to find the connections between the different elements of the data encoded by the EC.

Finally, the structure of the Hippocampus appears to be modular and that structure can be implemented using the Object Programming (OOP) paradigm. In this work, we assume the structure shown in Figure 3.1.

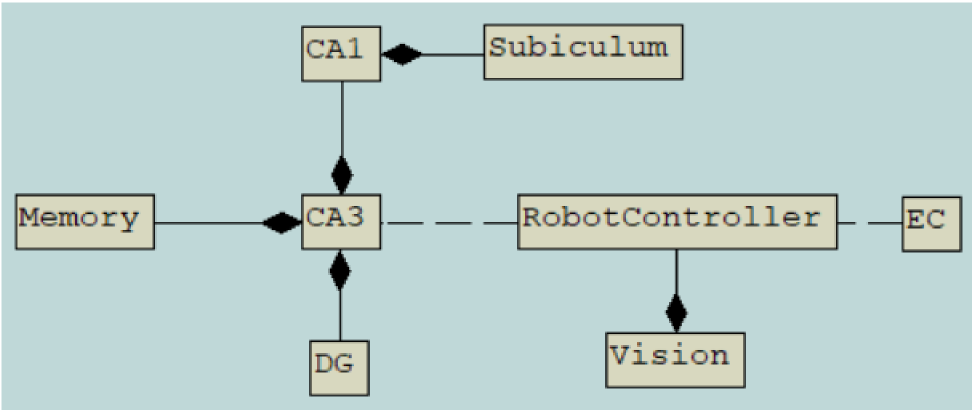


Figure 3.1: Proposed Model Structure of the Hippocampus and Entorhinal Cortex displayed in a Class Diagram Format

3.2 ENTORHINAL CORTEX

As mentioned in the section 2.2.3, the EC has a set of cells called Grid Cells, that are capable of marking a specific place in the environment. In this model, it is proposed that Grid Cells, implemented in the EC component, can identify places of interest with the help of a special group of Stripe Cells, as presented in the Self-Organized Model. Those Stripe Cells act as an encoding method of the area supported by a special group of coordinates that we call Triangular Coordinate System. As the name suggests, the Coordinate System is connected to a set of triangles with different sizes and each line of the triangle corresponds to a specific Stripe Cell. The Stripe Cell can represent multiple levels and it is the junction of these levels that creates the encoding system. The mentioned encoding system could encode different types of data with cyclic and non-cyclic properties. The usage of multiple areas allows the representation of different range values. However, this type of system requires a codification method that allows the identification of the different areas. A connection must be maintained when such areas are close to each other. The Gray Scale encoding method, defined in the following, allows such behaviour by defining closer transitions to neighbouring points.

The main goal, of the proposed model, is to achieve a method that recreates some of the Grid Cells behaviours without any complexed mathematical operations and with little use of external resources.

3.2.1 A GRAY CODE MODEL FOR THE ENCODING OF GRID CELLS

As stated in [64], the firing fields of Grid Cells: (1) show a repetitive equilateral triangular structure that (2) forms hexagons of equidistant firing peaks, at multiples of the distance to the nearest hexagon. The Figure 1 of [64] illustrates the mentioned characteristics. So, considering that Grid Cells encode the integration of the path, we consider that a triangular coordinate system must constitute the basis for that spatial encoding.

One of the requirements for that coordinate system is that the spatial indexing method should assure that two adjacent locations have a high degree of similarity. Also, the encoding system must be immune to transition errors when moving between adjacent locations. It should also allow the logical aggregation of locations to enable the indexing of assets and objects that might span over several spatial units.

Another requirement about the data indexing method is the ability to support periodicity. By looking the notion of time and space, it is possible to conclude that in some cases the data tends to repeat after certain periods. The analysis of variables that lead to that periodicity is very relevant. As the EC receives projections from the Subiculum, the periodicity of the Grid Cells will result from the periodicity of the representations of data at that point.

While in some cases the periodicity is clear, in other cases it is unclear if the observed data is going to have a repetition pattern. Therefore, the index should allow both the encoding of sequences that are periodic and non-periodic, of any length and of any definition.

The spatial encoding must take into consideration the different layers of Grid Cells, combined to get a unique encoding of each place in the environment. This means that for a given encoding, only one specific area will be associated. We consider that in the Hippocampus, the uniqueness of the encoding of a place is directly proportional to the uniqueness of the encoding coming from the EC and the remaining data that enters it. So, this requirement may be relaxed when considering the Hippocampal system as a whole.

Finally, the encoding should also be scalable, which means that the size of each encoded area and the coverage will depend on the number of layers. The proposed grid solution should be flexible enough to allow the coverage of small areas and, at the same time, scale to enable the coverage of large surfaces.

3.2.1.1 A TRIANGULAR COORDINATE SYSTEM

As stated in [65], the authors consider a recursive four-fold subdivision of a triangle, where the three edges of that triangle are bisected, producing four sub-triangles within the triangle. The method is repeated recursively, generating what they call recursion levels.

Similar to that method, in this section we specify a triangular coordinate system. However, we associate it with a ternary encoding. Let's start by considering an equilateral triangle, as the one shown in the top left image of Figure 3.2. Let's also consider that each of the vertices of that triangle is encoded with a ternary code (in this case 1, 2 and 3).

Starting at this point, a sequence of layers is going to be defined by a recurrent procedure that uses the previous layer to build a new one. The procedure is the following. At each of the layers, each of the edges of the triangles is bisected in two equal segments, creating a set of four smaller triangles inside the initial one, as shown in the top-right image of Figure 3.2. So, inside any triangle of a certain layer, three vertices are kept, and three new vertices are created.

Each of these new vertices will be encoded with the following method: given the ternary encoding (1, 2, 3 associated to any nodes A, B and C), the code given to a new vertex that result from the bisection of an edge between two vertices A and B will always be coded with a symbol different from A and B, thus being equal to the content of C.
For instance, if an edge connecting two vertices 2 and 3 is bisected, the newly created vertex will be encoded as 1. The same applies to any set of nodes to be bisected. The encoding of the other three vertices of the new layer is kept equal to the encoding of the previous layer. In the following, we will call this procedure the exclusivity rule, as it assures that a node encoded with a ternary symbol does not have a direct link to any other node with the same encoding.

As can be verified in Figure 3.2, as highlighted in images V and VI, the structure and associated encoding of all layers, form a grid of tessellating equilateral triangles, spanning through the whole of the original area.

The ternary encoding generates hexagons, where any direct neighbours of any node do not have a symbol equal to its own.

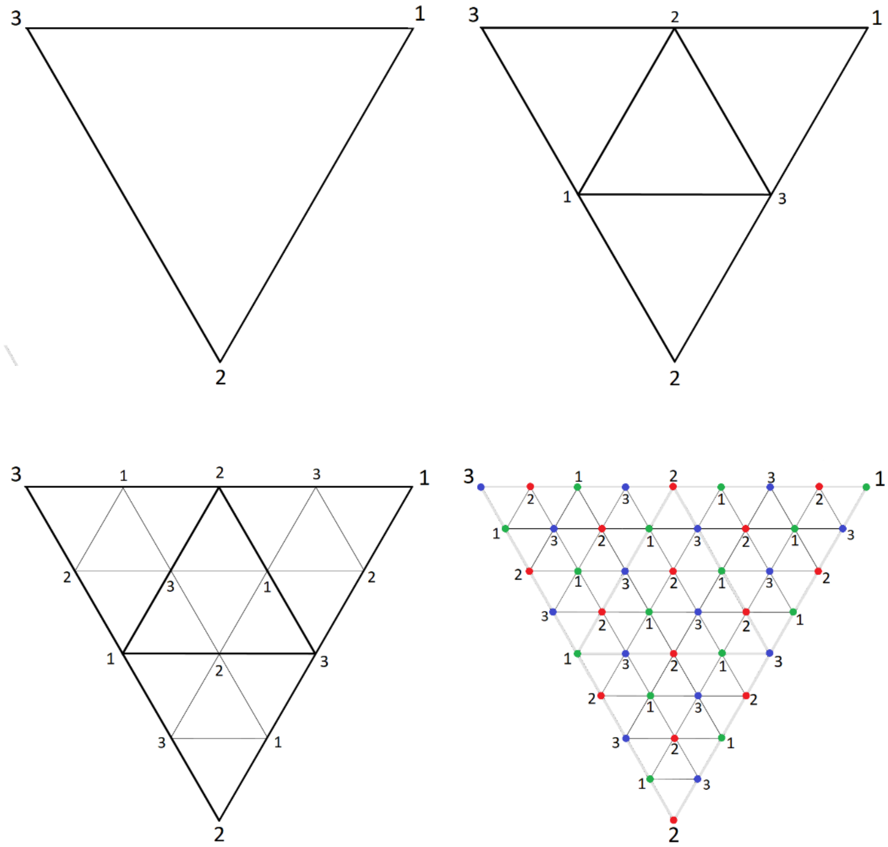


Figure 3.2: A triangular coordinate system for two-dimensional space: the upper left image represents the highest layer with 3 points; the upper right image represents the second layer with 6 points; the bottom left image represents the third layer with 45 points; the bottom left image represents the third layer with 45 points;

This subdivision process can be repeated any number of times. It is rep-tile. In each step, a new layer and grid are created, with smaller triangles. The number of points in each subdivision *i* can be obtained using the equation 1:

$$Points(i) = \frac{\left(2^i + 1\right)^2 + \left(2^i + 1\right)}{2} \tag{1}$$

where *i*, starting at 0, specifies the subdivision index.

This method enables the encoding of the points that belong to the vertices of each triangle in each layer but does not explain how to encode the points that stand inside each triangle. As in any quantification process, any point inside these triangles needs to be discretely encoded. In the following, we will propose a multilayer solution for that process.

3.2.1.2 ENCODING OF SPATIAL LOCATIONS

In this section, a space filling method is presented, that enables the encoding of the areas of each triangle that result from the fragmentation described in section 3.2.1.1. Starting from the highest layer, the process recurrently encodes several lower layers.

As verified previously, whenever a fragmentation of one triangle of layer L_i occurs, four lower-level triangles of layer L_{i-1} are created. To encode each of these four triangles, we are going to consider that a quaternary code is used, with the symbols 0, 1, 2 and 3. Figure 3.3 left side image represents that encoding. It works as follows: (1) each of the triangles that is closer to a vertex (as shown in Figure 3.2), assumes the code of that vertex, and (2) the triangle that is at the centre, equidistant from any of these vertices, assumes the value of 0 (or NULL).

For the encoding layer L_{i-2}, each of the triangles of the layer L_{i-1} is further fragmented in four smaller triangles. Again, each of the new triangles in Figure 3.3 that is closer to a vertex of Figure 3.2 assumes the value of that vertex, resulting in the representation shown in Figure 3.3 central image.

This process may be recurrently repeated any number of times. Figure 3.3 right image shows such a representation for the layer L_{i-2}. It can be verified that, as this process repeats, hexagons for the areas 1, 2 and 3 will be formed, as represented in the right image of Figure 3.3. These hexagons are organized in equilateral triangles and higher-level hexagons, when scaling up. Between them, NULL or 0 valued triangles will be placed.

This solution assures that each of the triangles of the lowest layer will be given a unique encoding. For instance, the encoding $[L_3; L_2; L_1] = [0; 3; 1]$ of the triangle represented in Figure 3.3 right side, is unique within the whole initial triangular area. It also assures that this process can be repeated any number of times, resulting in any 2D coverage and definition. The process is rep-tile.

The symbol 0 of the quaternary encoding, may either be represented as a function of the other three symbols or by the absence of an encoding. Although this latter option may seem uncommon, it is in fact used in digital communications, as for instance in the On Off Keying (OOK) modulation, a particular case of the Amplitude Shift Keying (ASK) modulation that does not send any signal when transmitting a logical 0.

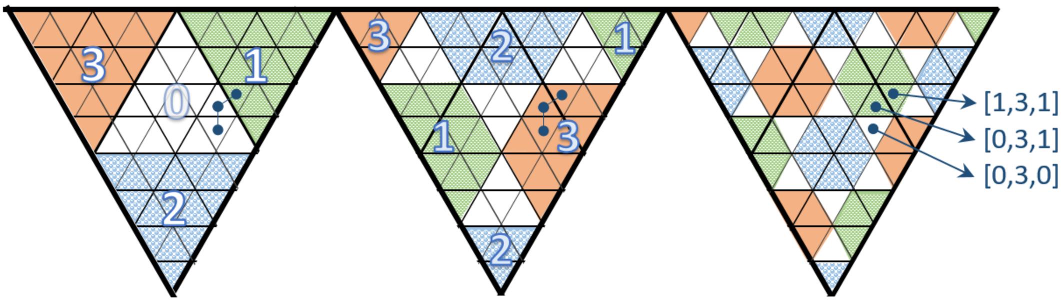


Figure 3.3: Discretization of the two-dimensional space using the proposed triangular coordinate system, where the left image represents the Top Layer (L_3), the central image represents the Medium Layer (L_2) and the right image represents the Lower Layer (L_1).

The complete Gray encoding is shown in Figure 3.4. It can be verified that each small triangle has a unique identification. The representation of each layer is made from the left to the right, meaning that the most left value corresponds to the highest layer and the rightest element corresponds to the lowest layer. If a new area is required, it can be obtained from the unfolding of this triangle using one of its edges. Then, the differentiation between these triangles is made using an upper layer, that needs to be added.

This solution assures that the transition between edges of the triangle results in a Hamming distance of 1. If the transition is made between vertices of opposite triangles, two ternary symbols will need to be changed. However, it can be verified that any of the codewords that stand in between these two codewords are always associated with neighbour triangles, and thus these temporary encodings will not be associated with distant locations.

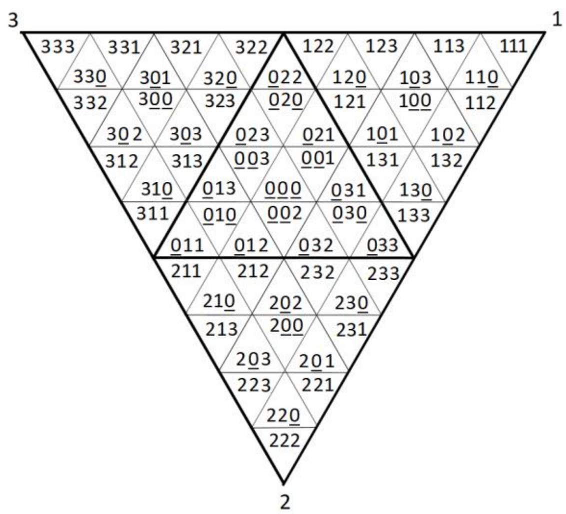


Figure 3.4: Resulting Gray code of the surface encoding

Let's now consider that, instead of using the quaternary code previously defined, we consider three binary variables A, B and C, that replace the encodings 1, 2 and 3 used in Figure 3.3. In this case, the 0 or NULL (white) area is represented as $(\bar{A} \cap \bar{B} \cap \bar{C})$ where the $\bar{A}$, $\bar{B}$ and $\bar{C}$ respectively represent the area that is not within the A, B or C triangles. The A, B and C variables are mutually exclusive, which means that any two of these variables cannot be active simultaneously. This same method enables obtaining the sub areas $(\bar{A} \cap \bar{B})$, $(\bar{B} \cap \bar{C})$ and $(\bar{A} \cap \bar{C})$.

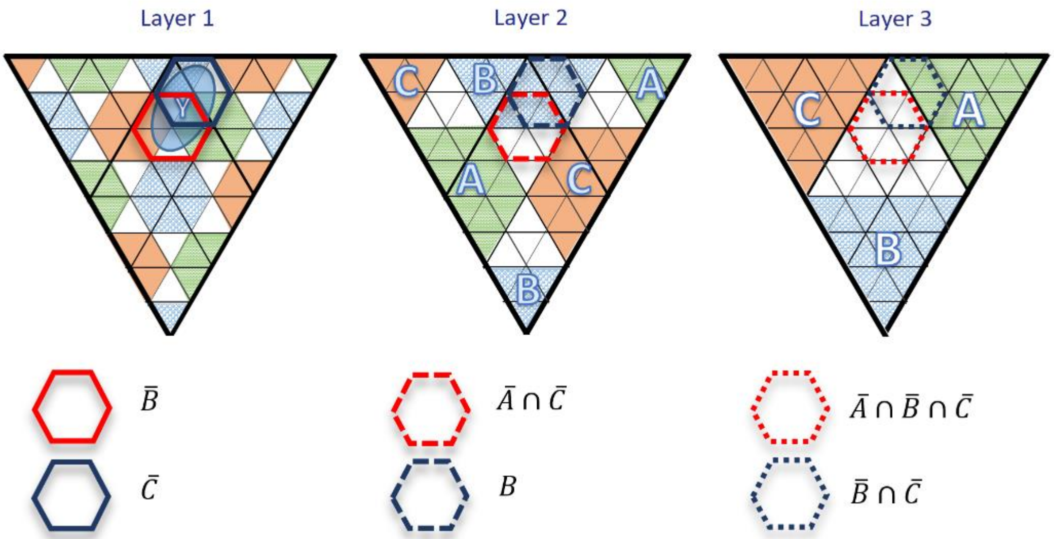


Figure 3.5: Aggregation of multiple areas using hexagons, represented from the lowest layer to the top layer.

One important feature that results from this representation is that it allows the aggregation of neighbour triangles, forming hexagons that simplify the logical conditions required to represent each of these areas. For instance, in order to encode/index the area Y shown in Figure 3.5 (left side), which may translate a spatial cue, two hexagons, red and blue, can be used. The red hexagon can be represented by $[L_3; L_2; L_1] = [\bar{A} \cap \bar{B} \cap \bar{C};\ \bar{A} \cap \bar{C};\ \bar{B}]$ and the blue hexagon can be represented by $[L_3; L_2; L_1] = [\bar{B} \cap \bar{C};\ B;\ \bar{C}]$. These two hexagonal aggregations replace the encoding of ten triangles, namely {[1; 2; 2], [0; 2; 2], [1; 2; 0], [0; 2; 0], [1; 2; 1], [0; 2; 3], [0; 2; 1], [0; 0; 3], [0; 0; 1], [0; 0; 0]} as represented in Figure 3.4. This process, that is performed from the lowest to the highest layers, significantly reduces the number of conditional detectors required to index areas. Also, as the two hexagons are partially overlapped, there is a continuity between the associated spatial representations, avoiding transition glitches/errors when moving between adjacent hexagons, inside an area. This process is similar with the one used in Karnaugh Maps to simplify real-world logic, so that it can be implemented using a minimum number of logic gates. The main difference is that, in this case, multilevel hexagonal aggregations are used, instead of the rectangular ones used in Karnaugh Maps.

This same process can be repeated in any higher layer, allowing the formation of larger hexagons that translate larger areas. For instance, a larger hexagon in the middle layer can be built using [L_3; L_2; L_1] = [$\bar{A} \cap \bar{B};\ \bar{C};$ x], where the x represents a "don't care" condition, i.e. any value is valid.

We hypothesize that these aggregations might be performed in the CA areas of the Hippocampus that receive the projections of the Grid Cells, leading to what is known as Place Cells. The aggregation process here proposed reduces the number of required conditional detectors required to index geospatial data that exists at different spatial scales, and thus takes into account the heterogeneity of the scales of that data.

After defining the two-dimensional Gray encoding, we now proceed to define one method that can be used to generate it.

3.2.1.3 A Gray Code Encoding for One-dimensional Maps

The purposes of this and the next sections is to define a method that allows the implementation of the 2D map specified in Figure 3.3. To achieve it, we start by defining a 1D Gray Code to be used in the scales of the three sides of the triangles shown in the last section.

In order to build these axes, let's consider a walk that goes through the boundary line of the equilateral triangle represented in the left side of Figure 3.3. Starting at the vertex 1 of this triangle and following clockwise the boundary line, the encoding that results from that walk is the one represented as Layer 4 in Figure 3.6. When this procedure is repeated for the triangle at the centre of Figure 3.3, the encoding that results from this procedure is the one shown in Layer 3 of Figure 3.6. The same procedure, applied to boundary of the right-side triangle of Figure 3.3, leads to the encoding shown in the Layer 2 of Figure 3.6.

Besides these three layers, more layers can be added in the top and bottom of this initial set, resulting in more codewords. This enables obtaining any predefined resolution and scale. These layers are represented as Layers 1, 5 and 6, in Figure 3.6. Only a part of Layers 5 and 6 are represented, as they span beyond the interval shown.

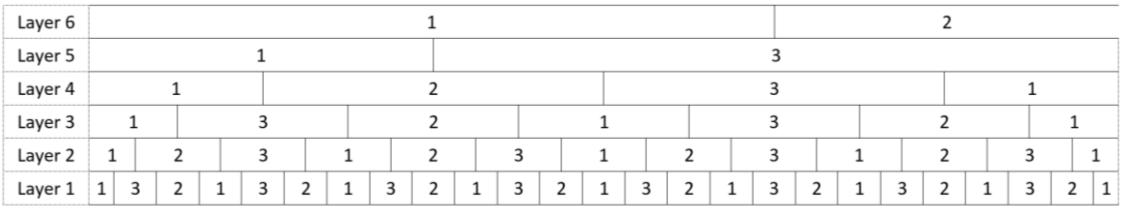


Figure 3.6: Non-circular encoding of the proposed Gray Code for a one-dimensional space.

The generation of this 1D Gray code can be performed by a set of neurons, each one having their own tuning curve. As described in the Neural Engineering Framework [66], distinct neurons can have different tuning curves, that characterize the sensibility each one has to a given input interval. The main characteristic of the encoding here proposed is that it has different layers of neurons, each one with three sets of neurons, that have tuning curves that repeat in a periodic and alternating way. Figure 3.7 represents the tunning curves used for the encoding process. Each colour represents a different area in the circular encoding.

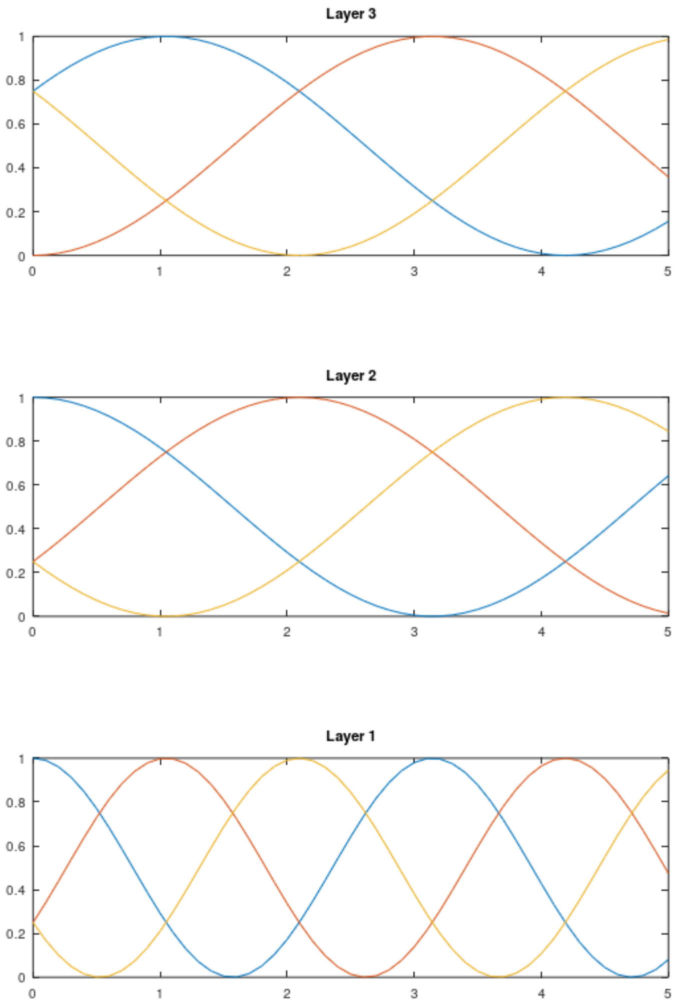


Figure 3.7: Tunning curves used for the development of the Gray encoding method.

As described, these patches are made of pyramidal neurons organized in a periodic and/or hexagonal pattern, whose dendrites bundle together and form a curvilinear matrix of structures, with a bulb-like [67] or tent-like [68] shape. We hypothesis that that bulb-like structure of pyramidal neurons can be used to generate an encoding similar with the one here proposed.

Different methods can be used to generate the encoding shown in Figure 3.6. As can be verified, the two upper layers (Layers 6 and 5) divide the input scale in three quantization intervals (QIs). Then Layer 4 subdivide each of these 3 QIs in two equal parts, thus introducing 3 partitions, creating a total of 6 QIs. Layer 3 adds 6 partitions, resulting in a total of 12 QIs. Layer 2 adds 12 more partitions, resulting in a total of 24 QIs. Finally, Layer 1 further fragments each of these layers in two, leading to a total of 48 QIs.

The defined encoding can be implemented using circular encoders, as shown in Figure 3.8.

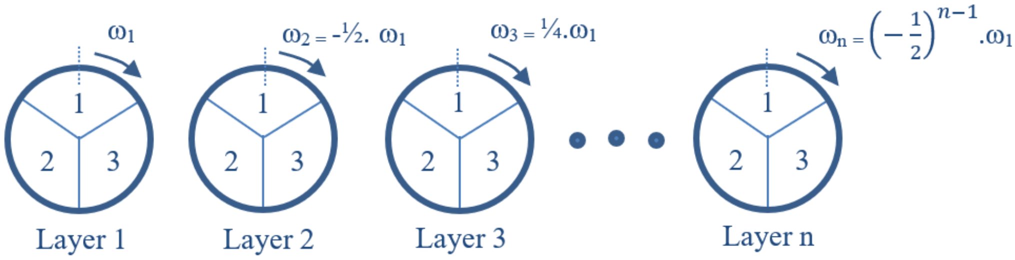


Figure 3.8: Generation of the Gray code using non-circular encoders. The dashed lines represent the initial phase.

The above-described procedures generate a Gray Code, as the one shown in Table 3.1. It can be easily verified that the Hamming distance between two consecutive encodings will always be equal to 1, assuring a high degree of similarity between consecutive encodings. The rotation relation between the different circular encoders is defined by the angular velocity (ω).

The Gray Code shown in Figure 3.6 is not closed. Thus, upper layers can always be added. The minimum number of layers should always be 2. For any number of layers NL≥ 2, the number of Qis (NQI) of non-circular code, can be obtained from the number of layers using equation 2:

$$NQI(NL) = 3 \times 2^{(NL-2)}, NL \geq 2 \quad (2)$$

Table 3.1: Non-circular Gray Code for a one-dimensional space

Decimal	Gray Code					
	L_6	L_5	L_4	L_3	L_2	L_1
0	1	1	1	1	1	1
1	1	1	1	1	1	3
2	1	1	1	1	2	3
3	1	1	1	1	2	2
4	1	1	1	3	2	2
5	1	1	1	3	2	1
6	1	1	1	3	3	1
7	1	1	1	3	3	3
8	1	1	2	3	3	3
9	1	1	2	3	3	2
10	1	1	2	3	1	2
11	1	1	2	3	1	1
12	1	1	2	2	1	1
13	1	1	2	2	1	3
14	1	1	2	2	2	3
15	1	1	2	2	2	2
16	1	3	2	2	2	2
17	1	3	2	2	2	1
18	1	3	2	2	3	1
19	1	3	2	2	3	3
20	1	3	2	1	3	3
21	1	3	2	1	3	2
22	1	3	2	1	1	2
23	1	3	2	1	1	1
24	1	3	3	1	1	1
25	1	3	3	1	1	3
26	1	3	3	1	2	3
27	1	3	3	1	2	2
28	1	3	3	3	2	2
29	1	3	3	3	2	1
30	1	3	3	3	3	1
31	1	3	3	3	3	3
32	2	3	3	3	3	3
33	2	3	3	3	3	2
34	2	3	3	3	1	2
35	2	3	3	3	1	1
36	2	3	3	2	1	1
37	2	3	3	2	1	3
38	2	3	3	2	2	3
39	2	3	3	2	2	2
40	2	3	1	2	2	2
41	2	3	1	2	2	1
42	2	3	1	2	3	1
43	2	3	1	2	3	3
44	2	3	1	1	3	3
45	2	3	1	1	3	2
46	2	3	1	1	1	2
47	2	3	1	1	1	1

In order to transform this non-cyclic encoding, into a cyclic encoding, the two upper layers must be replaced by one closed sequence, that violates the alignment (and the rule) that was previously used in the non-cyclic code to generate upper layers. An example of that procedure can be verified by observing the top layer of Figure 3.9.

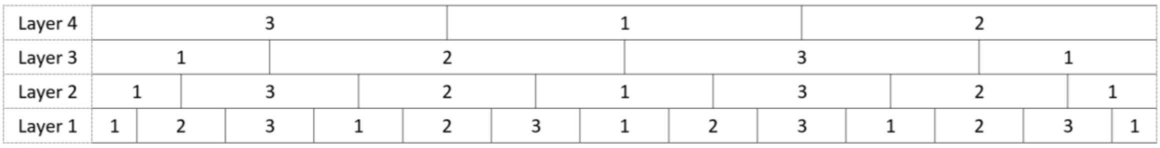


Figure 3.9: Circular encoding of the proposal Gray Code for a one-dimensional space generating 24 codewords

The generated Gray Code is shown in Table 3.2.

Table 3.2: Circular encoding of the proposed Gray Code for a one-dimensional space

Decimal	Gray Code			
	L_3	L_2	L_1	L_0
0	1	1	1	1
1	1	1	1	3
2	1	1	2	3
3	1	1	2	2
4	1	3	2	2
5	1	3	2	1
6	1	3	3	1
7	1	3	3	3
8	2	3	3	3
9	2	3	3	2
10	2	3	1	2
11	2	3	1	1

Decimal	Gray Code			
	L_3	L_2	L_1	L_0
12	2	2	1	1
13	2	2	1	3
14	2	2	2	3
15	2	2	2	2
16	3	2	2	2
17	3	2	2	1
18	3	2	3	1
19	3	2	3	3
20	3	1	3	3
21	3	1	3	2
22	3	1	1	2
23	3	1	1	1

Different from the Layers 5 and 6 of Figure 3.6, this upper layer closes the sequence, making it a circular, but at the same time a closed code. The precision may be increased by adding more layers at the bottom of the structure. This type of circular encoding is useful in many cases where periodicity is required.

For any number of layers NL$\geq$ 2, the number of QIs of the circular code can be obtained using the equation 3:

$$NQI(NL) = 3 \times 2^{(NL-1)}, NL \geq 2 \tag{3}$$

This circular sequence can be generated differently. An algorithm to do it can be the following, given a temporal sequence of periodicity T or an analog signal of amplitude T, divide the interval T in three equal intervals of length T/3 and encode each of them differently, using a ternary code. Then, for each of the three segments obtained in the first step, find a fragmentation point that subdivides them in two equal parts. This creates three fragmentation points, as seen in Layer 3 of Figure 3.9. Then, encode each of these intervals with the defined ternary encoding. Continue this procedure until a pre-required definition is achieved. The above defined encoding can also be implemented using circular encoders, as shown in Figure 3.10.

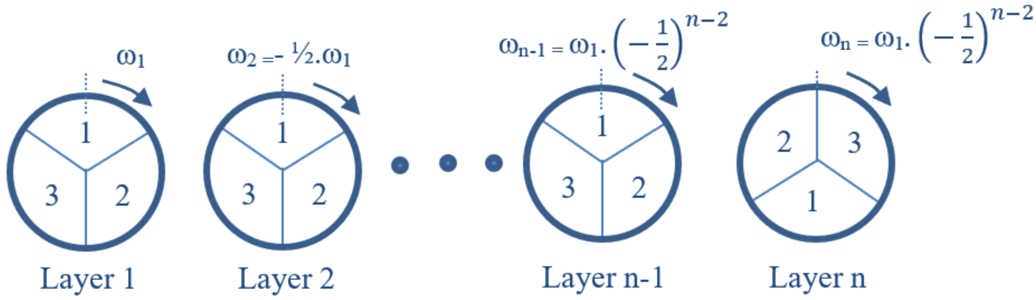


Figure 3.10: Generation of a circular Gray code using circular encoders. The dashed lines represent the initial phase.

3.2.1.4 NUMERICAL ENCODING FOR TWO-DIMENSIONAL SPACES

Given the Gray encoding of the three axes obtained in the section 3.2.1.3, we now proceed to define a method that enables the implementation of the two-dimensional encoding shown in Figure 3.3. The proposed method allows the generation of the Grid Cells' encoding for two-dimensional spaces, using a set of three circular encoders, as the ones defined in the last section. Figure 3.11 presents that method. As can be verified, the 1D encoding is applied circularly to the three axes of the equilateral triangle.

From these axes, three Grid Cells are defined: A, B and C, as shown in Figure 3.11. The encoding of each of these Grid Cells "A", "B" or "C" results from the overlap of axial codes with values of "1", "2" or "3". The triangles encoded with an "A" results from the overlap of the axial encodings (1, 3, 1), (2, 1, 2) and (3, 2, 3). The triangle encoded with a "B" results from the overlap of the axial encodings (3, 1, 1), (1, 2, 2) and (2, 3, 3). The triangle encoded with a "C" results from the overlap of the axial encodings (1, 1, 3), (2, 2, 1) and (3, 3, 2). All the other triangles are encoded as "0" or NULL.

The algorithm that enables the encoding of the Grids Cells is presented as Algorithm 3.1. Given two locations, the current and the last known positions, the algorithm uses as inputs the angular direction between these two points (represented in radians) and the associated distance. Using a cosine operation, these values are then decomposed in three individual contributions for the three axes, which are separated by 2π/3 angles.

These contributions then make the circular encoders rotate according to the associated amplitude and direction. Finally, the encoding of the Grid Cells result from the encodings presented above. All the remaining encodings are defined as NULL.

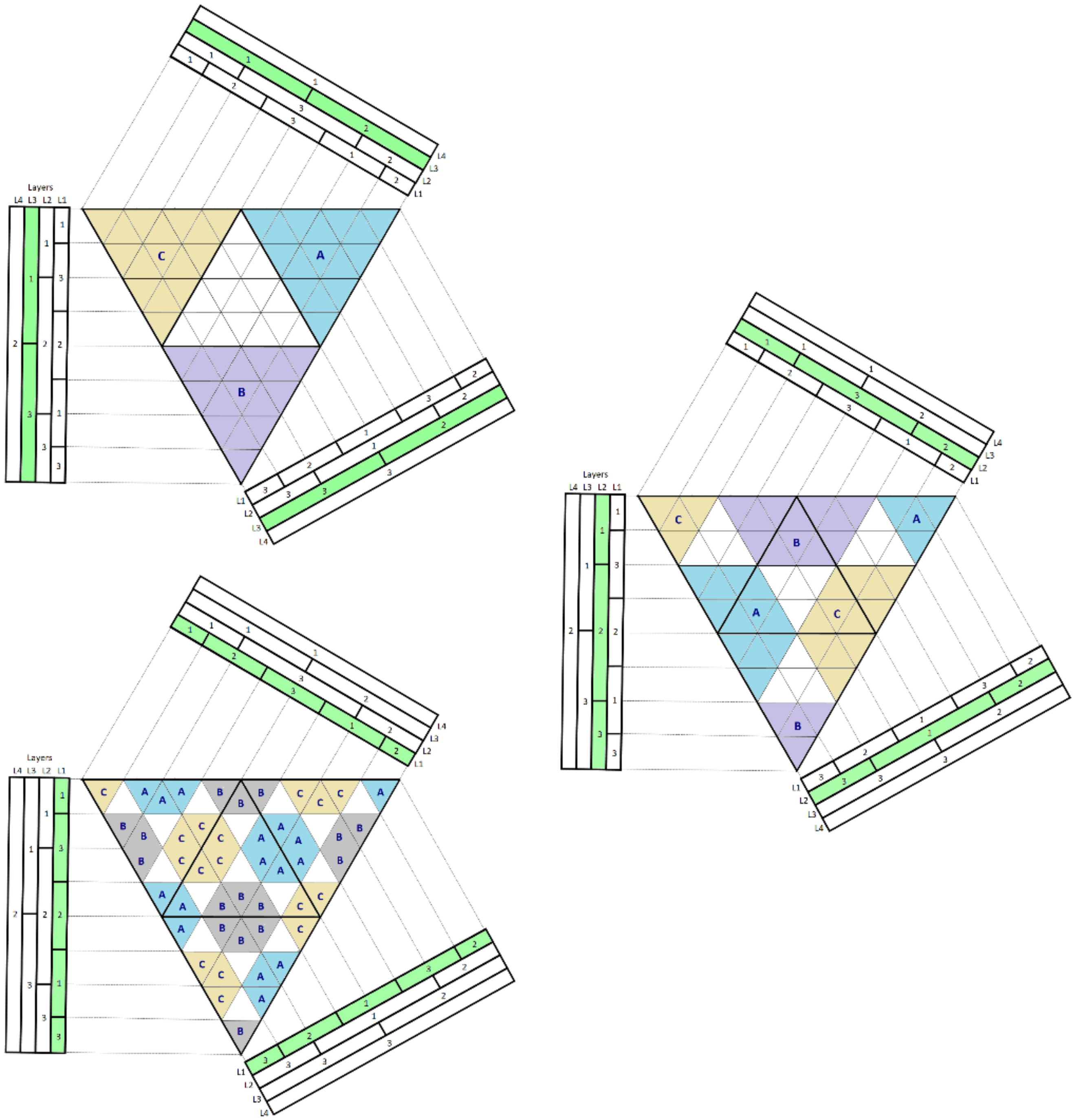


Figure 3.11: Encoding of three layers of Grid Cells using the three Gray Encoded axes.

The algorithm show as Algorithm 3.1 only encodes one layer. The encoding of the other layers requires adjusting the ratio between the distance travelled and the circular encoding as shown in Figure 3.8 and Figure 3.10.

Algorithm 3.1 – Algorithm for the computation of the Grid Cells in one Layer

Inputs: HD as the angular direction of the movement since last position, Distance as the distance from last position, AxesAngles as an array with the three angles of the axes and CircularAngles as an array with the current three angles of the circular encoders.
Internal: Coord as an array with the current three values of the axial coordinates
Output: GridCell as the Grid Cell encoding

```
AxesAngles      ←   [−π/6, π/2, 7π/6]
CircularAngles  ←   initial position for circular angles
                    for instance [0, 0, 4π/3]
foreach TimePeriod
    CircleCos ← cos([HD, HD, HD] – AxesAngles)
    CircularAngles ← (CircularAngles + Distance.*CircleCos) % 2π
    foreach pos in Coord
        if (π/3 < CircAngles(pos) ≤ π) then
            Coord(pos) ← 2
        else if (π < CircAngles(pos) ≤ 5π/3) then
            Coord(pos)  ← 3
        else
            Coord(pos)  ← 1
        end
    end
    if (Coord = [1,3,1] or Coord = [2,1,2] or Coord = [3,2,3]) then
        GridCell ← A
    else if (Coord = [3,1,1] or Coord = [1,2,2] or Coord = [2,3,3]) then
        GridCell ← B
    else if (Coord = [1,1,3] or Coord = [2,2,1] or Coord = [3,3,2]) then
        GridCell ← C
    else
        GridCell ← NULL
    end
end
```

Note: (1) the ".*" operator represents the element-by-element product between both vectors; (2) the "%" operator represents the modulus or remainder of the division.

3.3 A MODEL FOR THE HIPPOCAMPUS

As mentioned before, the Hippocampus is composed of different areas and those areas are the DG, CA3, CA1 and Subiculum. The Hippocampus can be seen as a memory indexer that allows the retrieval of previous stored memories, by comparing any new element with the previously memorized ones and is seen as an environmental map creator that identifies specific points of interest in a space. Each area has its own purpose and specifics. This will be further explained in the next subsections. One of the Hippocampus abilities is to learn new characteristics in real time, supporting a continuous learning process.

3.3.1 CA3

The CA3 area is one of the most important areas of the hippocampus. One of the main features of this area is associated with its recurrent connections. By observing Figure 2.3, there are in a rat approximately 12000 neurons responsible for the extraction and analysis of the information present in such connections. The CA3 area is also considered a pattern separator [69] [70] and it is assumed that distances calculators may be used to determine these data associations made by recurrent connections [71]. With such mechanism, the Hippocampus can identify events when the input data comes incomplete or with errors. Such behaviour can be achieved with the recurrent connections. With simple methods, it is possible to simulate the behaviour of a well-known ANN called Hopfield network [49]. The scheme presented in Figure 3.12 illustrates the hypothesis of a possible conversion from the CA3 biological model to the implemented model. Each of the recurrent connection of the CA3 area represents the state of an environmental variable. The association method can easily be achieved by calculating the distances or similarities between the current element and the memory elements that were previously captured.

The Euclidean [72] and the Manhattan [73] distances are used to compute the similarities, as shown in Figure 3.13. Those calculations allow the identification of the nearest event that corresponds to the input data and such metrics are used in different types of applications [74][75].

These mathematical operations are made in a specific set of elements. It is assumed that the CA3 has a set of event separators that fire when a specific condition is reached. The event separations were made using the lower and middle layers of the Grid Cells. When the encoded current data, obtained by the sensors, is equal to other existing elements, a similarity is detected. In case of doubts or if two results have a low distance between them, the associated elements are further separated by the DG module.

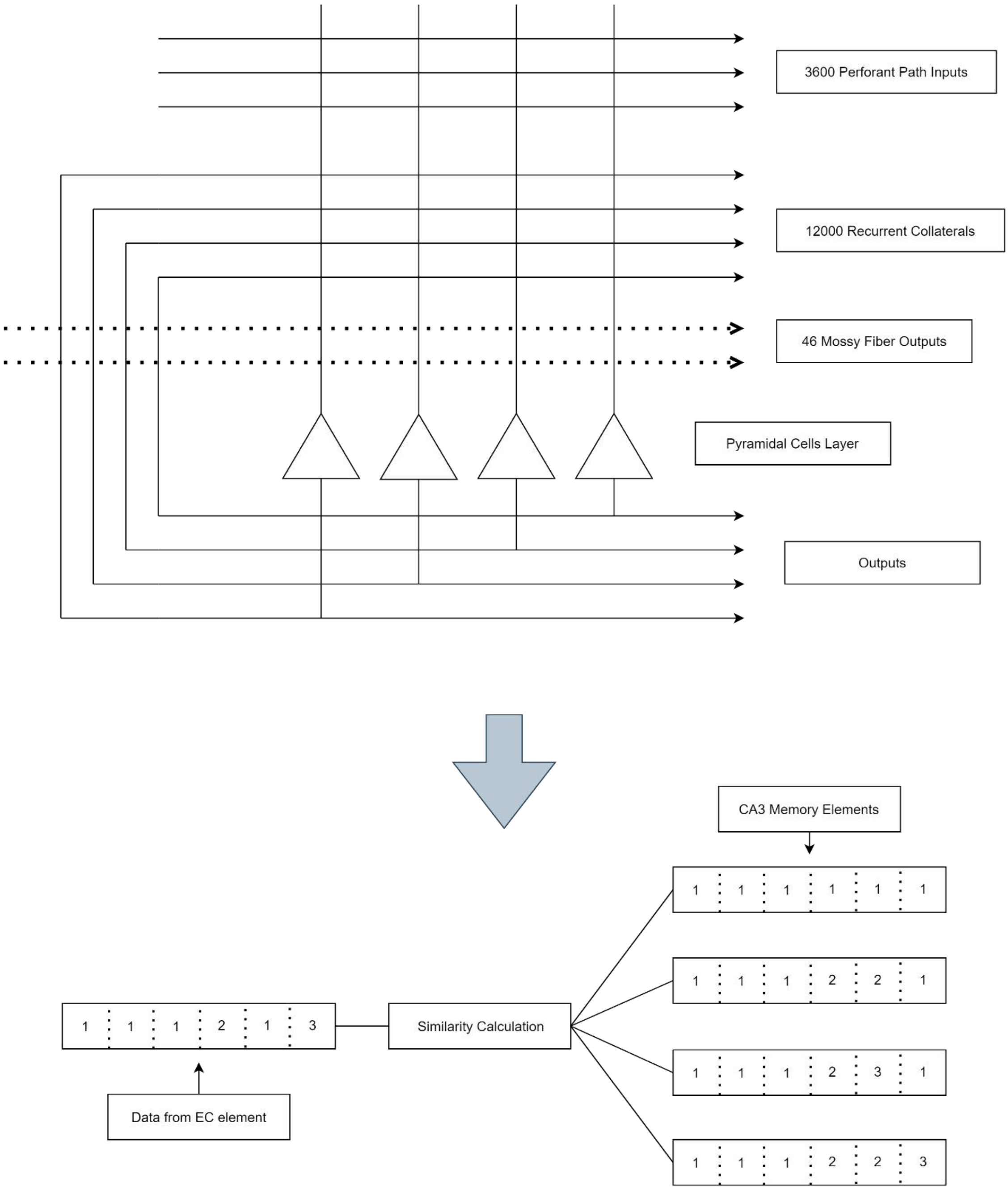


Figure 3.12: A possible conversion between the biological model (top side) and the proposed model (downside). This conversion tries to replicate the association and retrieval mechanism based on an occurred event and the elements stored in memory.

There is another system that acts as a complementary mechanism. This mechanism receives the CA3 result as an input and is executed after the CA3 module. The validation mechanism puts all the CA3 memory elements into a matrix and all those elements are directly compared with the current captured information. The similarity calculation is done using the dot product operation between the memory matrix, that contains all elements captured in a short-term and the current information represented in a transposed format. This type of calculation can be directly compared with the performance of Hopfield networks behaviour because both mechanisms are similar [49]. The result is the memory element that has the same index position of the highest value of the dot product result. A possible match between the result obtained from the pattern separation method and the dot product occurs when they agree in which memory element is the closest relative to the observed information. If there is a disagreement between the results from the CA3 and the complementary system, it is necessary to compare the obtained results with the received data encoded from the EC, by calculating the Hamming distance between the involved encoded elements.

After checking which elements are the most similar relative to the current observed data, the result is sent to the CA1 for the identification or creation of a grouped elements.

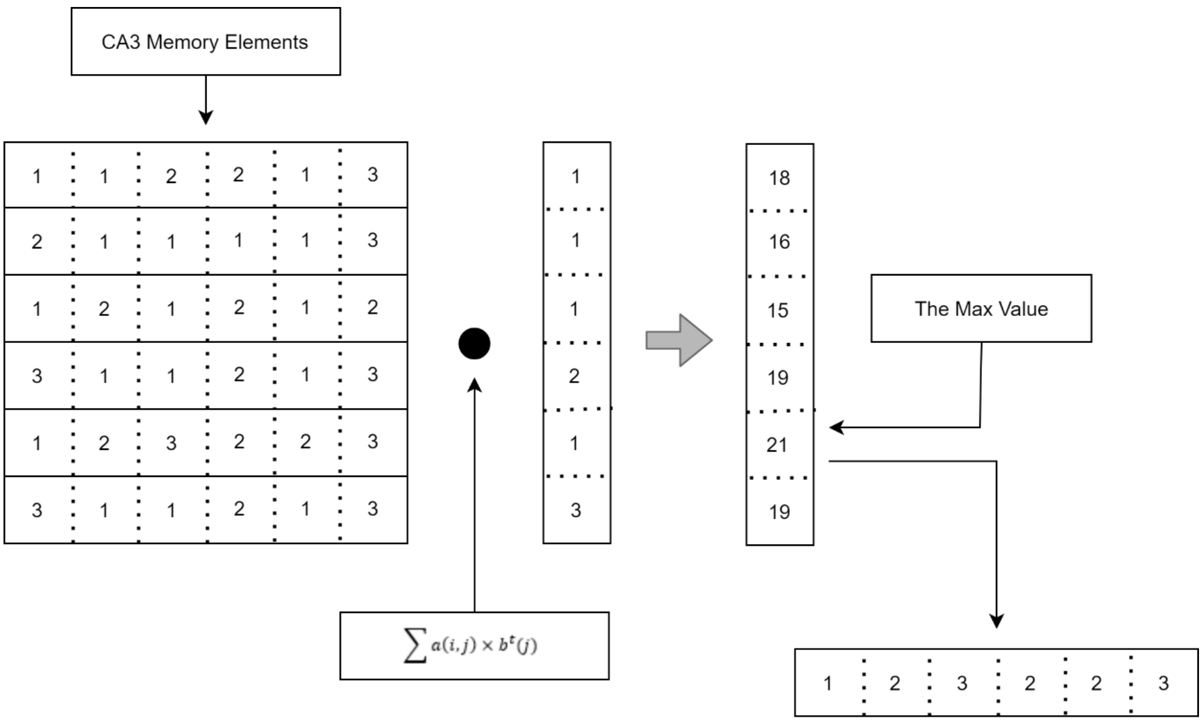


Figure 3.13: Complementary Similarity System. The matrix A represents the events stored in memory in a horizontal form and matrix B represents the occurred event.

By observing the Class Diagram in Figure 3.1, the CA3 module is connected to an auxiliary class called Memory. At the code level, this is implemented in a so-called Memory class that includes methods responsible for managing and controlling the elements that are inserted or removed. The Memory is divided into different parts and each part has a selector. The selector works like a group label, and its value are obtained from middle and lower layers of Grid Cells. The events that have the same middle and lower Grid Cell encoding will be inserted in the same section or memory part. This question will be further analysed in section 4.3.2, Figure 4.7 and in Figure A.8. The human brain does not have infinite resources and because of such characteristic it is necessary to remove the unused data. The forgetting behaviour can be described by a curve [76]. The curve is fitted using a mathematical expression called the Ebbinghaus Forgetting Curve and is described by the equation 4:

$$Q(t) = \frac{1.84}{\log(t)^{1.25} + 1.84} \tag{4}$$

where the t corresponds to the time in minutes [77].

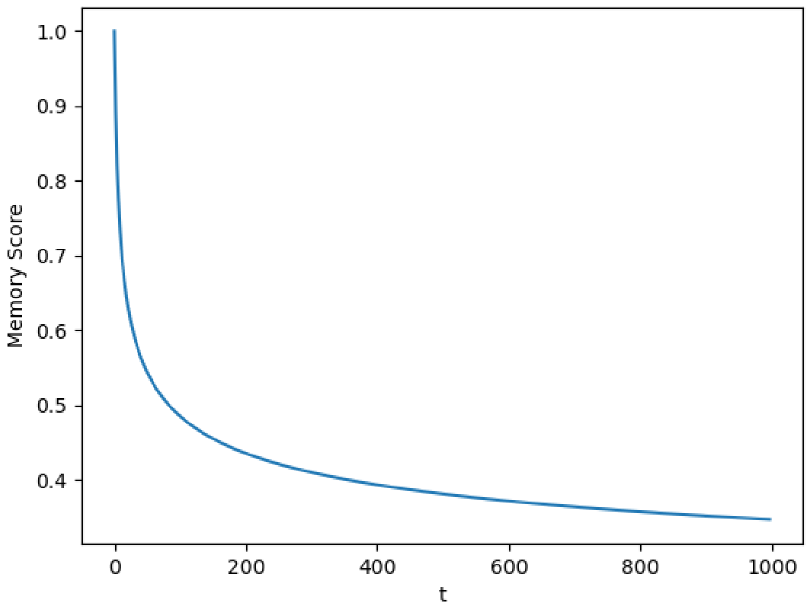


Figure 3.14: Ebbinghaus Forgetting Curve used by the implemented model

To simulate memory loss, t behaves like a counter. The memory scores decrease according to Figure 3.14. When the event reaches the lowest memory score, it is removed, but when a known event is observed, its memory score increases to the maximum possible value, allowing the identification of cycles and their cycle periods. This type of management allows the simulation of the brain's memory loss by removing events when they are not used for a given time.

3.3.2 Dentate Gyrus

In accordance with subchapter 2.2.1, the DG area can be seen as a pattern separator, that allows the identification of different elements in a sequence. The separation method tries to solve an ambiguity problem caused by similarities that might exist between elements. This module is only executed when there are elements with some level of ambiguity. Figure 3.15 shows the structure of the Dentate Gyrus module. Two different elements are inserted, the current data and the ambiguous memory elements. The distances of the different elements are calculated by using the Cosine Similarity formula and the Hamming distance. Such distances will determine the closest element relative to the received input. Both distances will be calculated and the minimum one is selected. If the result of both distances is the same, the Hamming distance value is selected. Such mechanism is possible if the result obtained from the Cosine Similarity method is inverted.

The Cosine Similarity measure calculates the angle between different elements by comparing their angles. Two similar elements have a smaller angle. This method can be found in different types of applications like the data mining operations [78].

The Hamming distance is the basis of many error detection techniques. This type of calculations directly compares the symbols that are different between two codewords [79]. For instance, if two codewords have a Hamming distance of 2, it means that 2 elements of these codewords do not share the same value at the same position. The simplicity of this method allows using it in telecommunications when data is affected by errors.

3.3.3 CA1

As mentioned in 2.2.4, the Place Cells are created in the CA1 area. It is proposed that the CA1 module must be able to identify specific characteristics based on the data provided from the CA3 module. By combining the result obtained by the pattern separation method made in DG and CA3 areas with the data obtained by the Entorhinal Cortex, the CA1 can identify a specific place of interest.

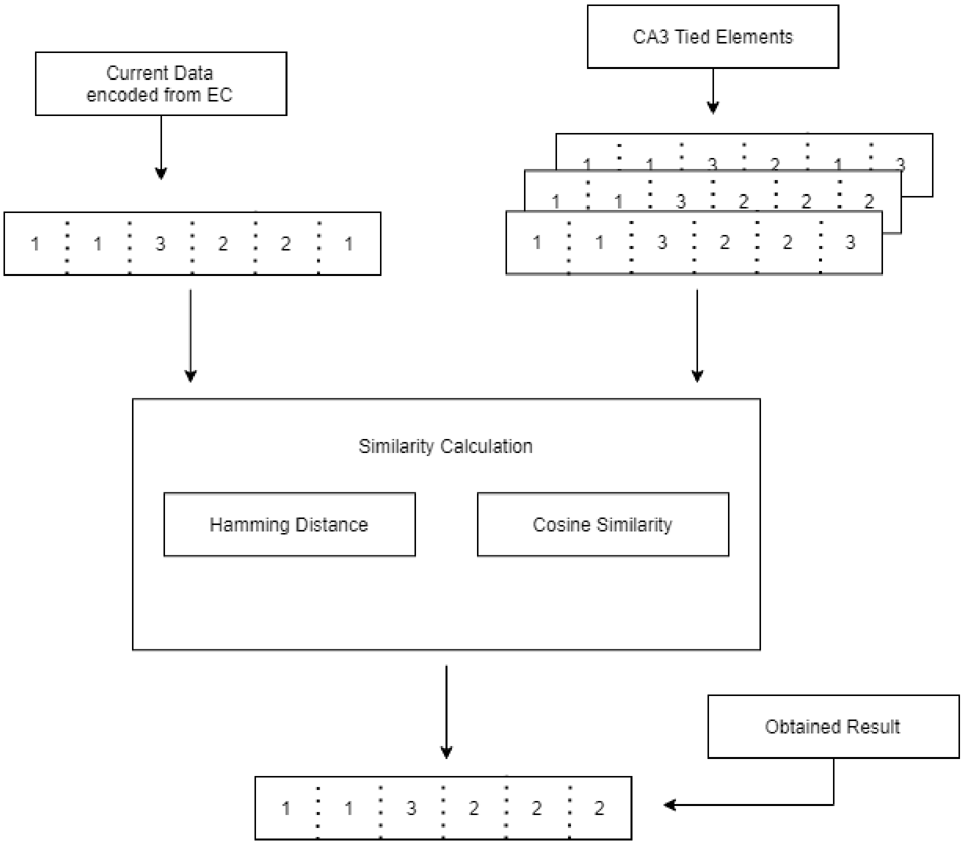


Figure 3.15: Dentate Gyrus disambiguate System

Like the CA3 area, the CA1 also has different types of separations and those separations are made with the elements obtained from the top layer of the Grid Cells.

Figure 3.16 represents the scheme of the Place Cells identification in the CA1. In this model, the Place Cells are created using two data sets, the information obtained from the sensors and the most similar element stored in CA3 memory. At first, the newest group element will start with both of elements. However, like the CA3, it is assumed that the CA1 area also has a set of selectors. In this case, the firing selector is the top layer of the Grid Cells. If the new set of elements have the same Grid Cell top layer codification, they will be added into the same Place Cell. The Place Cells are identified with sequential numbers and, unlike the CA3 memory, the Place Cells data will be permanently stored in memory. The Firing Connections in the Figure 3.16 represent the selection of the Place Cell where the obtained elements will be inserted.

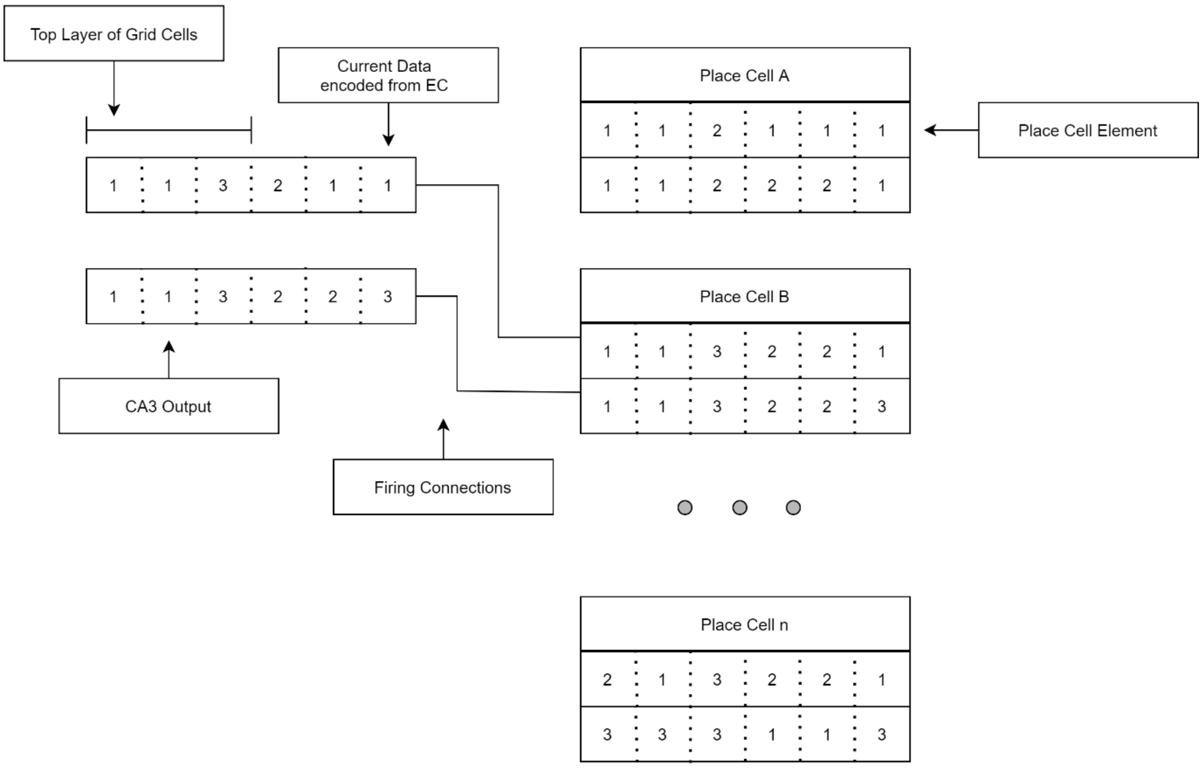


Figure 3.16: Scheme of the Place Cell Identification System

3.3.4 SUBICULUM

The consolidation of the obtained result in the CA1 area happens in the Subiculum. In the previously mentioned computational models, the Subiculum element is not given much attention. In terms of biological analysis, the Subiculum has HD cells and border cells, so it is possible that the Subiculum receives the information obtained from the CA1 and uses the HD and Border Cells to set a kind of self-awareness. In the model here proposed, the self-awareness method is simulated by finding possible connections between the identified Place Cell and the other known places. The calculation of such connections is made by the Q-Learning [61] algorithm. The Q-Learning method is a Reinforcement Learning algorithm that calculates a next state by giving the current action, making it suitable for navigation situations [61]. The states used for such calculation are the known Place Cells, calculated from the CA1 module, and the actions are the adjacent Place Cell extracted from the current state. The usage of Q-learning was a connection finder algorithm is justified by the information mentioned in section 2.3.3 that stated Hippocampus could play a role in the brain's Reinforcement Learning system.

Figure 3.17 illustrates how the Q-Learning is used to determine the associations between the identified Place Cell and the other cells. The Q-learning mechanism uses a rewarding table, and the table has information about the distances, between Place Cells. Those distances were calculated using the Euclidean Distance method. The distances calculated are used as a rewarding table. Then a Q-table is created, with the same dimensions as the rewarding table. It represents the possible connections available between each Place Cell. Initially, the Q-table is created with zeros, but they change when a value in a certain position is updated. An action from the rewarding table is randomly chosen from all possible actions and the Q-value is updated according to the equation 5:

$$Qvalue = rT[chosenState, chosenAction] + \gamma \times \max(Qtable[chosenAction]) \quad (5)$$

where the rT is the rewarding table and the γ is a selector factor with a range value between 0 and 1. The actualization of Q values requires several iterations and for simulating purposes the number of iterations and the γ are fixed in the model source code.

The Q-value updating equation was extracted from the article *"A Deterministic Improved Q-learning for Path Planning of a Mobile Robot"* [80]. According to [80], the used formula is considered an classical approach to Q-learning. This expression calculates the possible connections between Place Cells by updating the Q-values into the Q matrix using the previous calculated distances as a rewarding factor. The plot of the Q-table connections is represented using a graph network made by the Python Library called NetworkX, that can create complex network architectures with simple programming instructions [81].

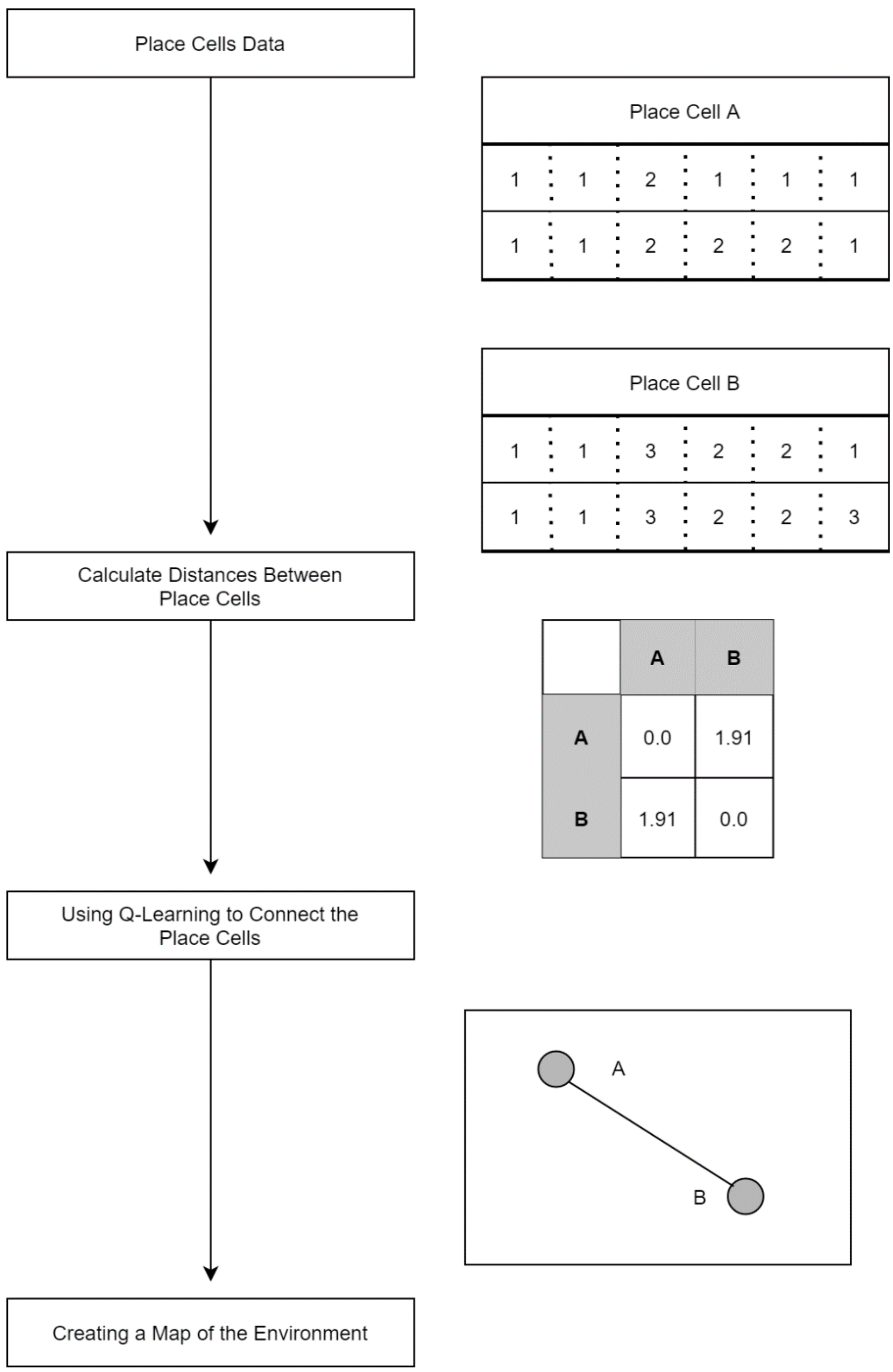


Figure 3.17: Proposed Model of the Subiculum

4. PERFORMED TESTS AND RESULTS

To validate the Grid Cells modules, the Entorhinal Cortex and the Hippocampus model, a set of tests were conducted. By testing the implemented model, it is possible to verify its coherence with the Biological model. The following sections will contain information about the testing conditions and it also illustrates several tests made on specific model methods. The final subchapter contains information about the Structural Test, which illustrates how the different elements should exchange information. The detailed information about the specific validation tests performed in specific modules can be found in the Appendix section.

4.1 TESTS CONDITIONS

The previous model was tested with the help of an open-source robot simulator called Webots. The software allows the creation of multiple environments and the usage of different types of robots [82]. The platform is very agile because it allows the implementation of the robot controllers using different types of programming languages. It is also possible to create or adapt robots by adding sensors and functionalities.

4.1.1 ROBOT ENVIRONMENT

The robot can move freely in a squared box with 2 by 2 meters. Each wall is painted with a specific colour to distinguish the different sections of the environment. The red wall is the

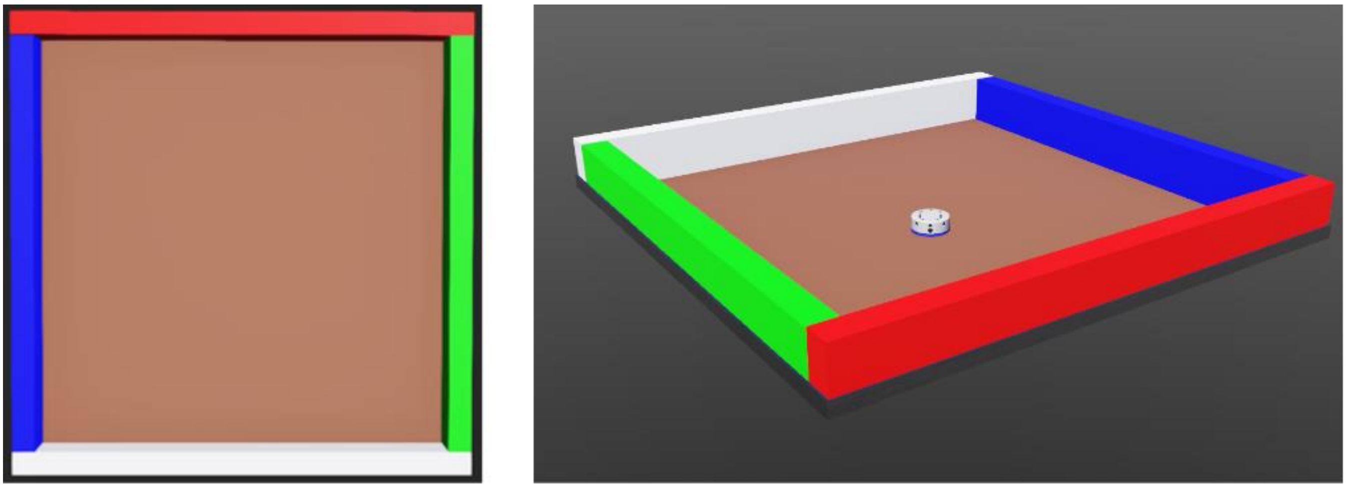


Figure 4.1: Simulated 2D and 3D Environment

North area of the environment, thus when the robot is facing that wall, the magnetometer will show the highest value.

4.1.2 ROBOT USED FOR TESTING

The simulations were made with a robot called the Khepera IV, developed by the K-Team [83]. It is a two-wheel platform with a set of distance sensors. It is equipped with an accelerometer, gyroscope, and camera. Because the Khepera IV is not equipped with a magnetometer, to find the course direction the robot configuration was changed to include a magnetometer. The magnetometer sensor is a direction-finding sensor that reaches the highest value when an individual is facing to the magnetic north. With that information, it is possible to find the direction of a subject. However, it is still required to extract the bearing or the direction of movement from the magnetometer values. The conversion between the magnetometer values to direction in degrees is calculated by the equation 6 [84]:

$$Direction = \frac{[\mathrm{atan2}(MagnetometerX, MagnetometerZ)] - \frac{\pi}{2}}{180 \times \pi} \tag{6}$$

4.1.3 ROBOT CONTROLLER

The robot controller was made with Python programming language. The Python language is a high-level language. It has many supporting libraries and allows to keep the focus in the implementation of the algorithms by avoiding the technical aspects. The robot controller allows two types of movements: a forward movement, which consists in applying the same velocity to both wheels, and a rotating movement which applies opposite velocities. The robot rotates itself over its own axis, by applying a forward velocity to one wheel and a negative velocity to the other. The rotations can be done to the left or to the right. The movement within the environment can be made in two ways: a straightforward movement or a random movement. The velocity is the same throughout the simulation. The Head Direction is obtained by the magnetometer value and the Border Cells value is obtained by the infrared Distance Sensors. They can detect environmental obstacles or walls when a certain value is reached. If a wall is detected, the robot starts the rotation to the left until the obstacle is clear.

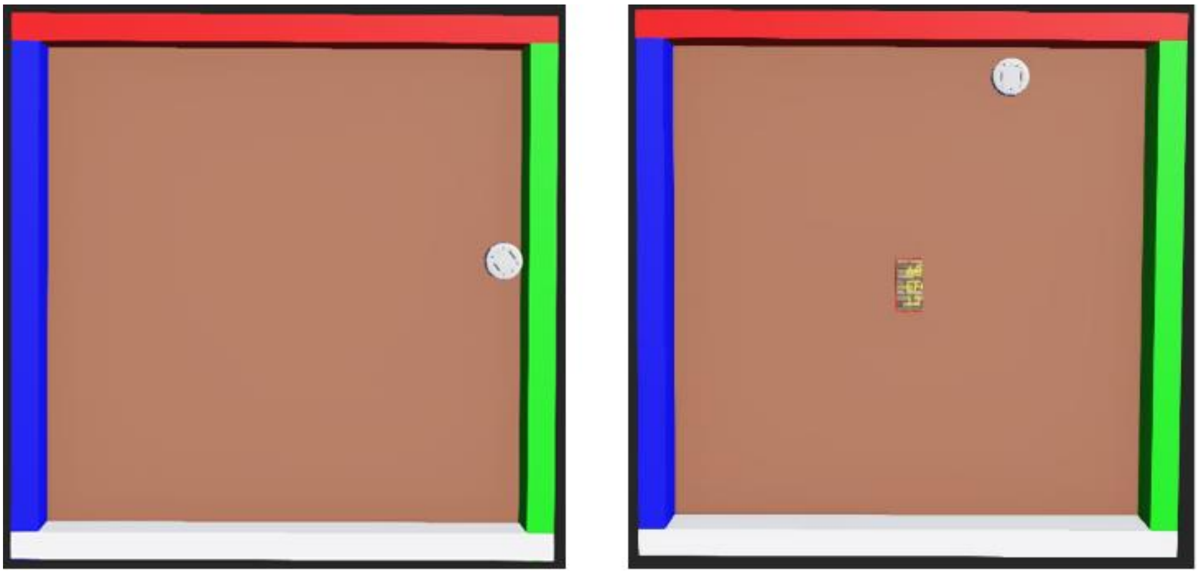


Figure 4.2: Robot's environment without obstacles (left side) and robot's environment with obstacles

4.1.4 VISUAL DATA

By observing Figure 3.1, it can be verified that the RobotController has an association with a class named Vision. The Vision class is responsible for the extraction of features from the frames captured by the camera's device. The purpose of such component is to verify if the proposed model can work with different types of complexed data like the extraction of visual information. The feature extraction from the frame is made with the Sobel Operator [85]. The Sobel Operation is an edge detector algorithm used for computer vision applications. Applied to an image, the algorithm can introduce a smoothing effect on the random noise and enhances the edges by calculating the orthogonal gradient [85]. The Sobel method uses a mask that is directly convolved with the image. The mask can have multiple sizes and the result of the convolution is an image with the edges enhanced [85]. The usage of the Sobel operator is adequate for finding possible obstacles and environmental edges that can be used to decide the firing of a possible Place Cell. However, as mentioned before, the implemented model does not interpret the frame in its format but as a number. The number is calculated by separating the initial frame into four different areas, called quadrants, and the result is the sum of the pixels average for each quadrant. Figure 4.3 represents an example of the mechanism implemented in the Vision component.

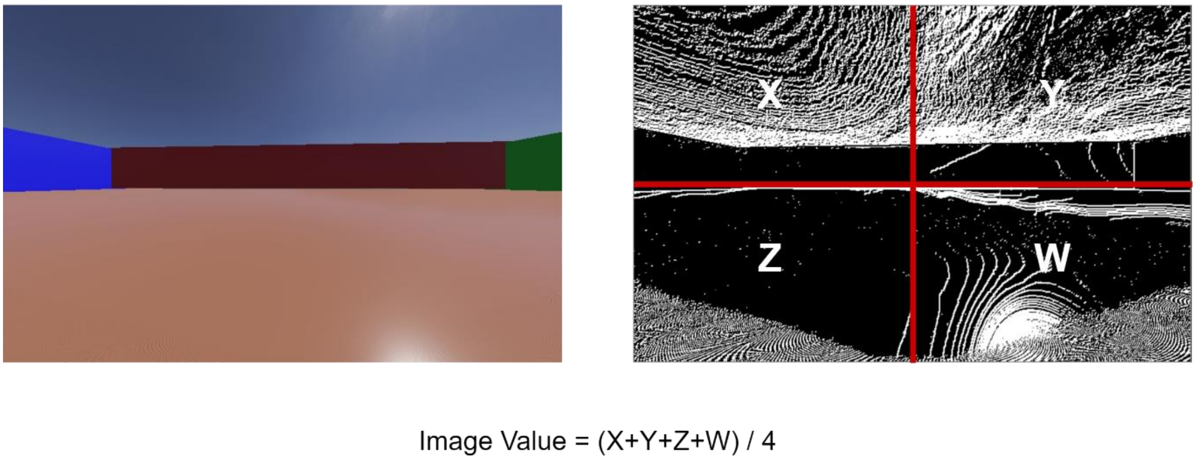


Figure 4.3: Example of the Vision class behaviour

4.2 TESTING OF THE ENTORHINAL CORTEX MODEL

In this section, we focus on the testing of the model for the Entorhinal Cortex to verify if the proposed Grid Cell model is an efficient way to map an environment. As described in section 3.2, the Entorhinal Cortex model contains the necessary calculations for the creation of simulated Grid Cells.

The Grid Cells mechanism is implemented in the Entorhinal Cortex component. The mechanism for their creation is described in Algorithm 3.1. However, in its functioning, there are some constant parameters. In the following tests, the chosen angle's directions were $-\frac{\pi}{6}$, $\frac{\pi}{2}$, and $\frac{7\pi}{6}$. These angles define the places where the circular encoder is partitioned in 3 codes. The implemented Grid Cells are drawn in three different layers and each layer is encoded with the ternary encoding method described in section 3.2.1. To facilitate the visualization, each set of layer values has a unique symbol and colour, marking each area distinct. The velocity used by the algorithm was obtained by dividing the velocity measured by the robot by a constant (660). This value allowed the adjustment of the scale of the Grid Cells to the size of the space that is being mapped.

In Figure 4.4 top image, it is possible to verify that the Grid Cells model generates the expected multilayer hexagonal encoding of the space.

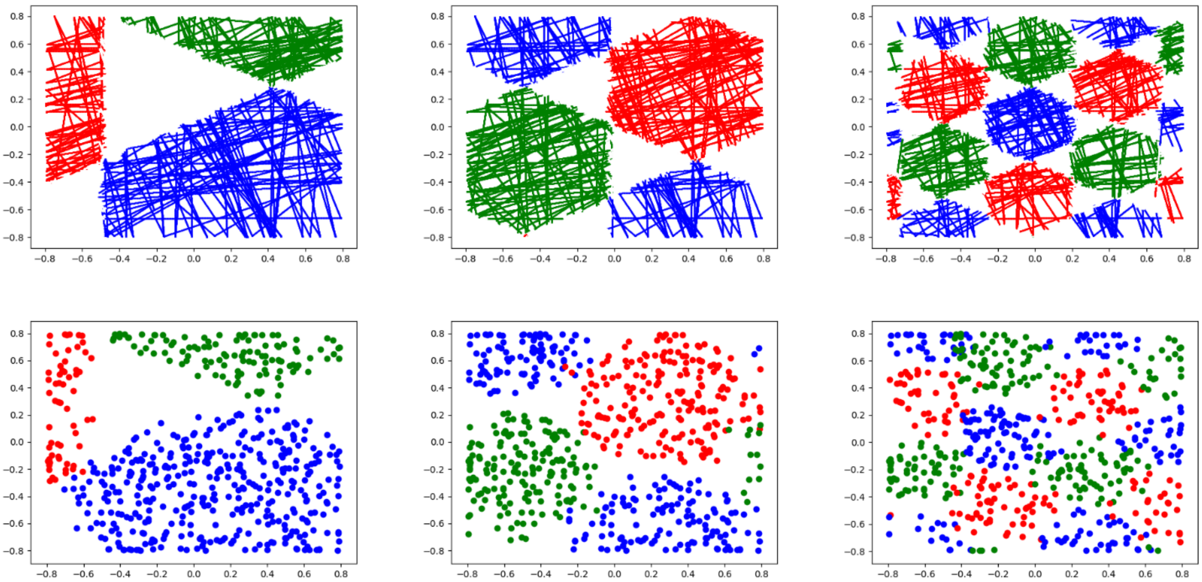


Figure 4.4: Grid Cells results in Webots environment without error (top side) and with a ±3.5% error on Head Direction data (down side)

This image represents the different layers that were calculated by applying Algorithm 3.1 to the robot movement.

The bottom images of Figure 4.4 were obtained using the same algorithm but forcing a ±3.5% error in the readings obtained from the Magnetometer. The error was generated using a uniform distribution. As can be verified, the error in the magnetometer causes the grid structure and hexagonal shapes to become less well defined. Grid cells become rounder. Besides that, the error does not compromise the structure, neither within each of the layers, nor between distinct layers. The three axial triangular structure tends to compensate the Magnetometer errors that do not accumulate, nor cause persistent drifts.

The proposed algorithm is also suitable for mapping environments with obstacles. By simulating the robot with an obstacle in the middle of the space, the area around the obstacle is correctly mapped as shown in Figure 4.5. The white square in the centre of this image corresponds to the obstacle, as the robot did not circulate in the associated area. The same behaviour is replicated into the different layers.

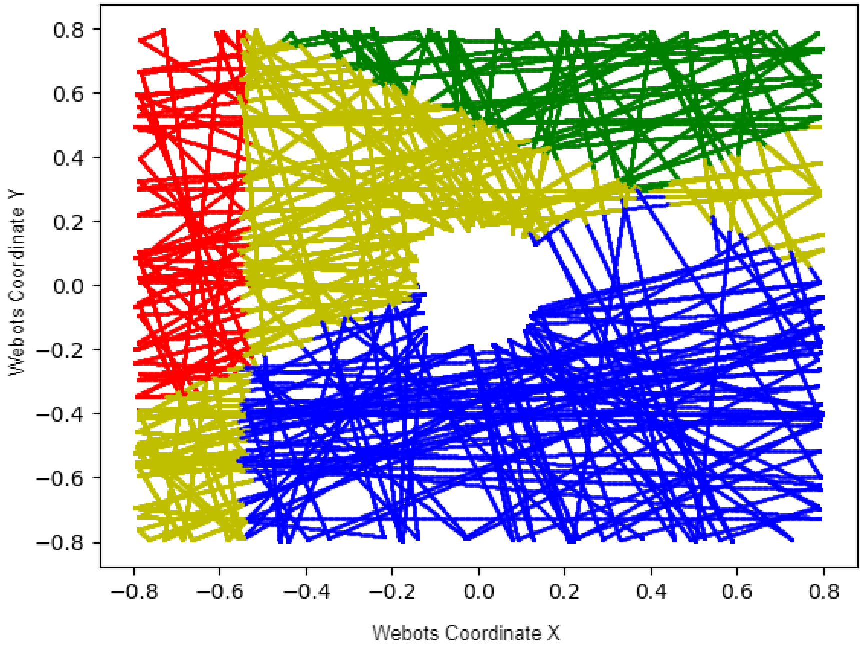


Figure 4.5: Grid Cells Top Layer Result with Obstacle

The results show that the implemented grid model is an efficient way to map an environment without the usage of big mathematical calculations or complex sensors

4.3 Testing of the Hippocampus Model

The Hippocampus model is composed of different types of modules, as mentioned in section 3.3. Every module has a unique set of features and constraints that are required to be addressed. In the following subsections, it will be described the parameters and results of the modules.

4.3.1 Encoding of the Head Direction, Sensor Data and Camera Feed

In the Entorhinal Cortex model, described in the 3.2.1.3, it is indicated that various types of data can be encoded in similar way as Grid Cells. Despite their increase in size relative to the raw data, the transformation of numbers into different ternary bits could facilitate the calculation and identification of some characteristics and similarities. The main objective is to verify if the encoding system is flexible enough to encode different types of data.

In the tests performed in the following sections, the data received from the Head Direction, the readings of the distance sensors and the result of the feature extraction from the camera feed were all encoded with the procedure described in the 3.2.1.3.

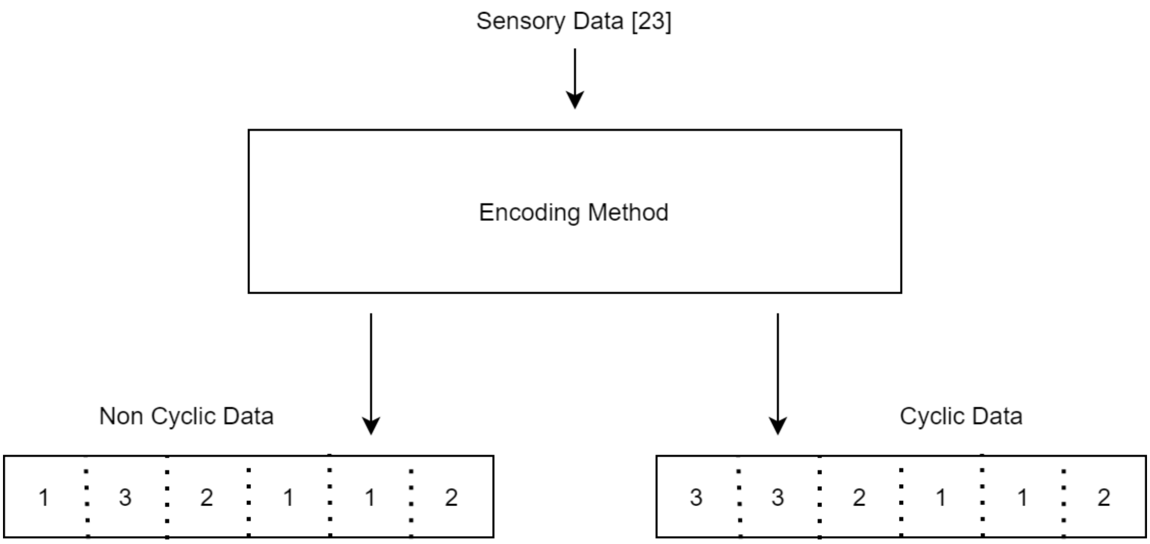


Figure 4.6: Example of the Encoding process of sensory data for the value 23

Figure 4.6 shows the implemented structure of the encoding process. The encoding follows the procedure described in section 3.2.1.3. As described, the codeword is obtained using several circular encoders, each one divided in 3 distinct areas. The number of circular encoders is equal to the number of layers. The methods for non-cyclic and cyclic encodings are similar and the main difference is in the top layer. When it is necessary to close the loop, obtaining a circular encoding, the two upper layers must have the same angular velocity, but use a different partitioning of the circle as shown in section 3.2.1.3.

4.3.2 Testing of the CA3 and Dentate Gyrus Models

The CA3 and Dentate Gyrus module is implemented according to the model described in 3.3.1 and in 3.3.2. The illustrated tests allow the verification of the association methods used in such components. Figure 4.7 illustrates the implemented functioning of CA3. The middle and lower layers of the Grid Cells from EC are used to create a memory area selector. The similarities are calculated using the Euclidean and Manhattan distances. Then, the two lowest distances are selected and the difference between such values is calculated. If the difference is equal to zero or below 0.25, these elements are sent to the DG module to be further analysed.

The distances of the DG module are calculated with a Python library called Scipy. In this library the Cosine Similarity result is inverted, meaning that the result of equal elements is 0 [86]. If the Hamming distance and Cosine Similarity return the same result, the value calculated by the Hamming distance is selected.

The CA3 module has direct interconnections with the Memory module. This module has a set of features that allows the removal of events not recently used. During the testing phase, an event not reseen during 10 iterations is considered obsolete and removed from memory.

The results illustrated in Figure 4.7 shows the identification of a place by comparing the current element provided by the EC module and the stored elements on the Memory component. The current and the selected elements are sent to the CA1 module.

The data elements used during the association test were obtained with the Webots robot simulator when the robot made 546 steps in an open environment illustrated in Figure 4.1. No obstacles were used during the data extraction, velocity was maintained, and the Head Direction value was extracted directly from the magnetometer value.

4.3.3 TESTING OF THE CA1

As mentioned in the 3.3.3, the CA1 module receives two inputs: the current observed data by the robot, encoded via EC, and the most similar memory element outputted by the CA3 module. Using such elements, the CA1 must be able to identify unique places in a space. The identification and creation of the Place Cells occurs inside CA1. The Place Cells selectors are originated by the information obtained from the top layer of Grid Cells.

In Figure 4.8, the results of the functioning of CA1 module are represented. For plotting reasons, each Place Cell has a unique colour. According to section 2.2.4, the CA1 area must have the ability to identify areas of interest by using the CA3 output as a reference. The obtained areas are defined and delimited, even when an obstacle is placed in the middle of the environment. By observing Figure 4.8, it seems that the CA1 its immune to obstacles. By comparing Figure 4.8 with Figure 4.5, it can be verified that the encoding of the areas is similar. That behaviour can be explained by the fact that the chosen selector in Place Cells is the top layer of Grid Cells and, because of that, each Place Cell uses the selector information to set their Place Fields. The decision was based in the fact that in biological terms there is not much evidence about the size and dimensions of the biologically created Place Fields. The set of the different layers of Grid Cells plays a major role in the place field displacement. However, there are other factors that contribute to the recognition of the Place Cells, like the data from the camera and the information that comes from the distance sensors.

4.3.4 TESTING OF THE SUBICULUM

The subiculum module uses the Place Cells data gathered from the CA1 element. Its main objective is to find any associations between Place Cells. Those associations would form a conceptual map of the different areas of interest. In the following tests, the Q-learning algorithm is responsible for finding the potential connections between different encoded elements by the EC module. As mentioned in 3.3.4, the Q-learning equation has a specific set of parameters. During the test phase, the Q-learning was set to find connections during 12 iterations with a selector parameter of 0,8. In each iteration a state is randomly selected, and a Q-value is calculated according to the method described in 3.3.4.

Figure 4.9 represents the result of the Subiculum module. Each node of the graph presented in Figure 4.9 represents one area interpreted by the CA1 module (shown in Figure 4.8). Figure 4.10 shows the associations between these Place Cell’s areas and the identification number shown in Figure 4.9.

As can be verified in Figure 4.9, the area numbered as 2 is directly connected to areas 1, 5 and 0. In Figure 4.10, these are in fact neighbouring areas of 2.

However, according to Figure 4.9, the Place Cells 4 and 2 are neighbours, but in Figure 4.10 they are not directly connected. Such behaviour may occur when the Q-learning system does not have the necessary iterations to solve all possible connections between Place Cells, because Q-learning works by a trial-and-error method. It is also possible that the chosen learning parameters may be not the best for these types of problems. Finding the best learning parameters and alternating the number of iterations may solve these incoherences. This analysis will be considered in future work.

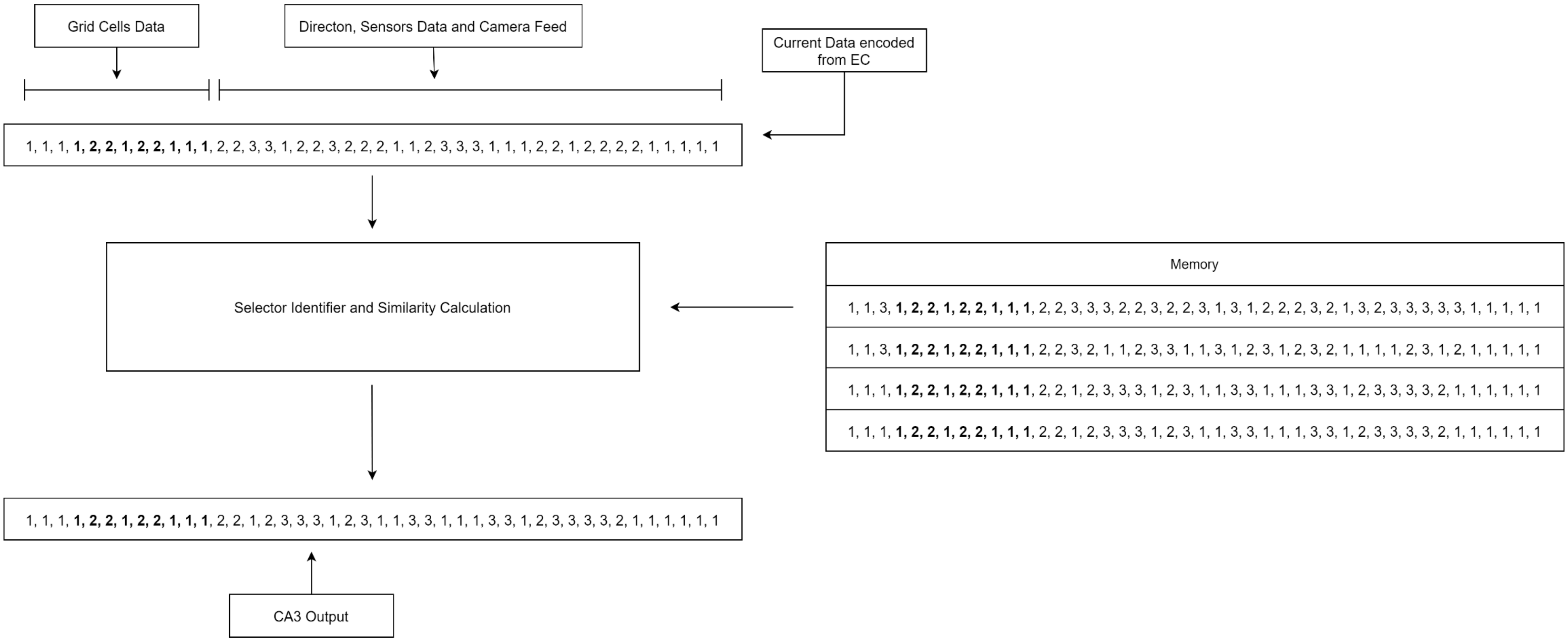


Figure 4.7: CA3 Results during Test Phase

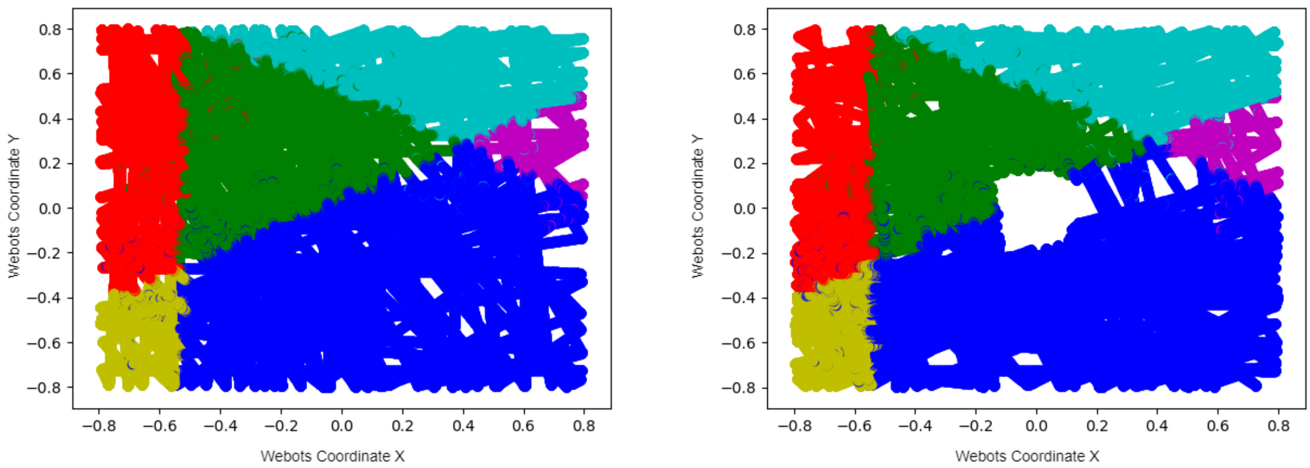


Figure 4.8: Area Displacement of Place Cells Firing occurring in the CA1 module without obstacles (left side) and with obstacles (right side)

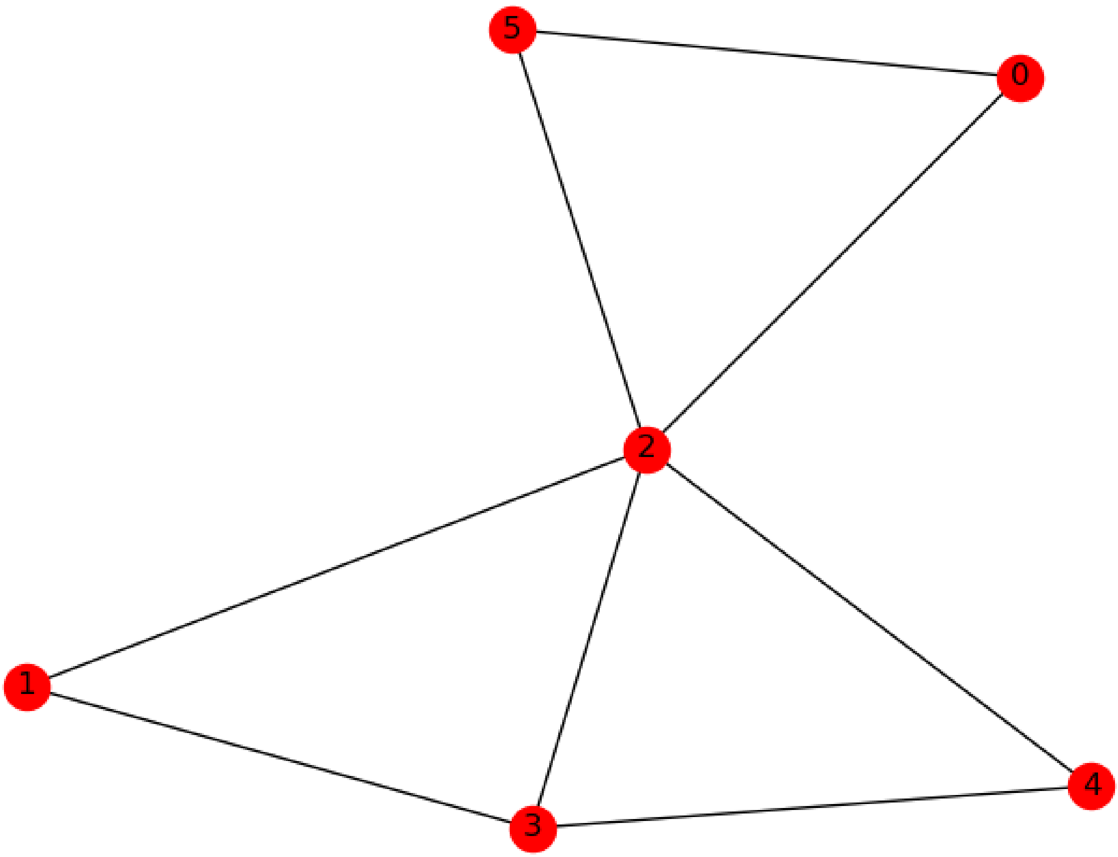


Figure 4.9: A possible topological connection between Place Cells calculated by the Subiculum module. The connections shown in this figure represent how the different places are connected. They could be connected by the Grid Cells data or the other sensorial data.

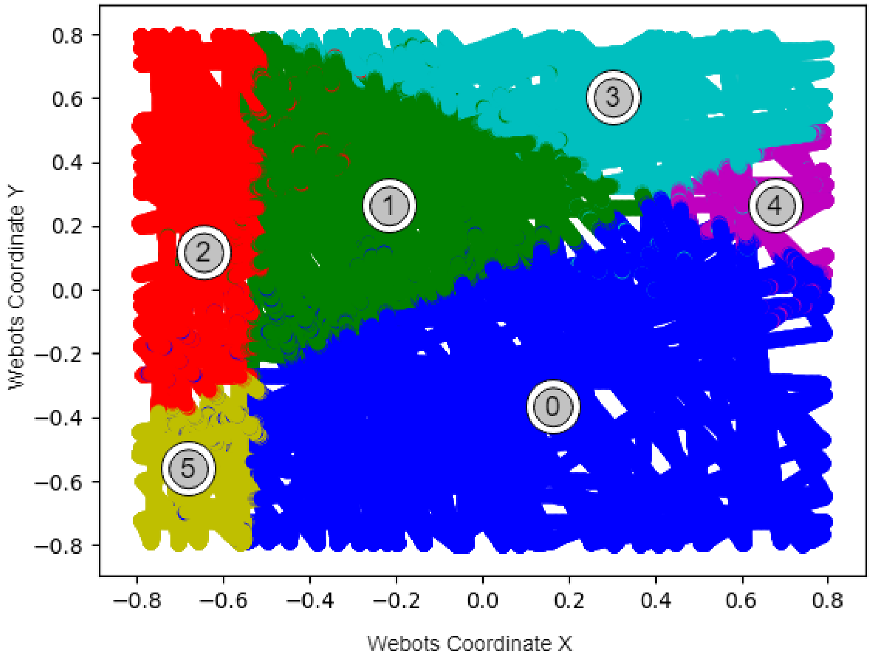


Figure 4.10: The enumeration of the different areas calculated by CA1 module

5. CONCLUSIONS AND FUTURE WORK

This dissertation focused in two major areas: (1) the Biological area in which the biological structure, connections and interactions between the Entorhinal Cortex and Hippocampus elements were explained and (2) the Computational part that explores the computational mechanisms and models used to simulate some of the biological functions occurring inside the Temporal Lobe of animals.

According section 2, neurons inside the Hippocampus, including Pyramidal neurons, are crucial for event recognition and allow storage of the newly acquired information. The scientific community has no doubt that the Hippocampus is crucial to memory formation, but no one knew about its navigational capabilities until the exploration of Grid Cells and Place Cells. It seems also now clear that the Hippocampus can organize events that occurred in a specific time frame because Place Cells can also encode time elements [36].

As mentioned in section 2.3.3, in the Reinforcement Learning field, the trial-and-error method is used to get environmental cues and some authors suggested that Hippocampus could play a major role in the brain Reinforcement Learning system. It could be also a Continuous Learning element because the Hippocampus has the ability to separate the tasks into several smaller events. It can learn such events without any training data or initial settings.

When comparing the module described in section 3 with the biological model, some simplifications were made. For instance, the EC only connects with the CA areas and the Subiculum does not return anything back to the EC. Also, no neuronal networks were used. In the EC module, there is an abstractedness in where specific data components are originated.

On the other hand, the Grid Cells were successfully created without the usage of complex mechanisms like the Oscillatory Interference Model or the Ring Attractor Network. As mentioned in section 2.3.2, those models require complex mathematical operations and neural networks to achieve the same functionality. With the proposed model it is possible to achieve similar results with a lower amount of inputs and mathematical operations. Also, the implemented Grid Cells model, only works in the positive domain, it can encode different parts of a space using 3 symbols and, according to the results presented in section 4.2, the model can tolerate small amounts of error. The direction of movement and the velocity are the key elements essential to map entire environments by using a multilayer system. The proposed encoding method can be used with different types of data and can represent the same information in a ternary way. Such feature allows the representation of different types of data into a triangular coordinate system and the proposed method, described in 3.2.1.1, could be useful in other types of applications.

When it comes to the Hippocampus, several implementational constrains and modifications to the biological data were made. One of them is the simplification of the connections in the DG and CA3 areas. In these modules, the similarity and untie methods focus on the entire encoded data. They do not look individually into specific sensory data. As mentioned in sections 3.3.2 and 4.3.2, the DG module was only required when there was a possible mismatch or a set of data elements highly similar. The implemented Memory module was created considering memory storage constrains. By eliminating the last referenced element, it is possible to avoid the memory overflow.

However, the same behaviour does not occur inside the Place Cells. When an element is removed from the Memory, the event is kept inside the Place Cells because it is believed that Place Cells memory can keep their data for months [87]. If the dataset is large enough, the memory requirements could scale inside the CA1 module. Another characteristic regarding the CA1 module is aggregation. The module is capable of clustering information previously calculated in the CA3. Yet, in this case, a question arises about which aggregation method should be used. It is not clear if the aggregation method should be specific with certain information, or if this method should consider that the similarity calculators are made for each variable and not in the entire encoded information. The aggregation method must be efficient to represent multiple areas and use the least number of cells.

The Subiculum is an element of the Hippocampus that require more biological studies. It is composed of different types of cells like the Head Direction and Border Cells. In the proposed model, described in section 3.3.4, it is assumed that the Subiculum can create some kind of map that connects the different Place Cells. This could be important when our brain needs to fully form a memory. The Place Cells map would allow the recollection of different elements or events associated to the memory episode. To improve and approximate the Subiculum module to the biological form, it is still necessary to further research about its interaction with the EC. Also, to study if the Subiculum can communicate directly with other brain elements.

The implemented models focus on the functionality of the different elements and how they can be combined to produce a valid output. The results show that there is some consistency between models and biological data.

While one of the main goals of this dissertation was to research create a computational model of the hippocampal formation that could be used to identify unique places in an environment, with the right configuration, the model can be adapted to other types of variables, including rhythm detection, anomaly management, recollection of specific events and event prevision.

With the rapid increase of IoT devices that can capture various types of aspects like the temperature, humidity, energy consumption, solar radiation, among others, the usage of such system allows the analysis of events that occurred in a specific timeline with the ability to analyse different types of data displaced in a multidimensional manner. Such characteristics are essential to find patterns and optimize mechanisms that use the environment and its interactions as a learning factor.

5.1 FUTURE WORK

The area explored during this dissertation is an emergent one and can have multiple applications into different areas like self-navigation systems, anomaly detection, event categorization and recognition.

Despite the large amount of work in the biological interactions that occur inside the brain, there are still many unsolved questions about the functioning of some elements. Therefore, it is necessary to further explore how these elements work and how they can be transformed into a computational system.

The implemented models of the Entorhinal Cortex approximate in a way, the biological Hippocampus and the EC elements. However, more work needs to be done to further improve them. It is necessary to study different approaches to this problem using for instance ANN or other types of methods.

The main idea is to create and/or improve existing models to make them biologically closer to the real interactions that exist between the Hippocampus and EC, while assuring that they are able to run on devices with restricted processing and memory capabilities.

5.2 PUBLICATIONS

The work done in this thesis supports a pending submission of an article and a patent request, both with the title:

- "A Gray Code for the Encoding of Grid Cells in the Entorhinal Cortex" (Submission Pending)

BIBLIOGRAPHY AND REFERENCES

[1]W. B. SCOVILLE and B. MILNER, "Loss of recent memory after bilateral hippocampal lesions.," *J. Neurol. Neurosurg. Psychiatry*, 1957.

[2]D. C. Rowland, Y. Roudi, M.-B. Moser, and E. I. Moser, "Ten Years of Grid Cells," *Annu. Rev. Neurosci.*, 2016.

[3]T. J. Teyler and P. DiScenna, "The Hippocampal Memory Indexing Theory," *Behav. Neurosci.*, 1986.

[4]T. J. Teyler and J. W. Rudy, "The hippocampal indexing theory and episodic memory: Updating the index," *Hippocampus*. 2007.

[5]C. M. Bird and N. Burgess, "The hippocampus and memory: Insights from spatial processing," *Nature Reviews Neuroscience*. 2008.

[6]E. Tulving and H. J. Markowitsch, "Episodic and declarative memory: Role of the hippocampus," *Hippocampus*. 1998.

[7]M. Moscovitch and L. Nadel, "Consolidation and the hippocampal complex revisited: In defense of the multiple-trace model," *Current Opinion in Neurobiology*. 1998.

[8]C. M. Bird and N. Burgess, "The Hippocampus Supports Recognition Memory for Familiar Words but Not Unfamiliar Faces," *Curr. Biol.*, 2008.

[9]A. Konkel and N. J. Cohen, "Relational memory and the hippocampus: Representations and methods," *Frontiers in Neuroscience*. 2009.

[10] A. D. Ekstrom and C. Ranganath, "Space, time, and episodic memory: The hippocampus is all over the cognitive map," *Hippocampus*. 2018.

[11] T. Bus, "Genetic investigations into the role of ionotropic glutamate receptors in hippocampal learning." .

[12] E. Cherubini and R. Miles, "The CA3 region of the hippocampus: how is it? What is it for? How does it do it?," *Front. Cell. Neurosci.*, vol. 9, Feb. 2015.

[13] A. Farovik, L. M. Dupont, and H. Eichenbaum, "Distinct roles for dorsal CA3 and CA1 in memory for sequential nonspatial events," *Learn. Mem.*, 2010.

[14] N. Spruston, "Pyramidal neurons: Dendritic structure and synaptic integration," *Nature Reviews Neuroscience*. 2008.

[15] S. Poulter, T. Hartley, and C. Lever, "The Neurobiology of Mammalian Navigation," *Current Biology*. 2018.

[16] E. T. Rolls, “A quantitative theory of the functions of the hippocampal CA3 network in memory,” *Front. Cell. Neurosci.*, 2013.

[17] E. T. Rolls, “An attractor network in the hippocampus: Theory and neurophysiology,” *Learning and Memory*. 2007.

[18] S. M. Dudek, G. M. Alexander, and S. Farris, “Rediscovering area CA2: Unique properties and functions,” *Nature Reviews Neuroscience*. 2016.

[19] I. H. Brivanlou, J. L. M. Dantzker, C. F. Stevens, and E. M. Callaway, “Topographic specificity of functional connections from hippocampal CA3 to CA1,” *Proc. Natl. Acad. Sci. U. S. A.*, 2004.

[20] R. S. Sloviter and T. Lømo, “Updating the lamellar hypothesis of hippocampal organization,” *Frontiers in Neural Circuits*. 2012.

[21] S. M. O’Mara, S. Commins, M. Anderson, and J. Gigg, “The subiculum: A review of form, physiology and function,” *Progress in Neurobiology*. 2001.

[22] M. Postans *et al.*, “Uncovering a Role for the Dorsal Hippocampal Commissure in Recognition Memory,” *Cereb. Cortex*, 2020.

[23] T. J. Voneida, R. M. Vardaris, S. E. Fish, and C. T. Reiheld, “The origin of the hippocampal commissure in the rat,” *Anat. Rec.*, vol. 201, no. 1, pp. 91–103, Sep. 1981.

[24] F. Attneave, M. B., and D. O. Hebb, “The Organization of Behavior; A Neuropsychological Theory,” *Am. J. Psychol.*, 1950.

[25] C. Keysers and V. Gazzola, “Hebbian learning and predictive mirror neurons for actions, sensations and emotions,” *Philos. Trans. R. Soc. B Biol. Sci.*, 2014.

[26] E. V. Lubenov and A. G. Siapas, “Hippocampal theta oscillations are travelling waves,” *Nature*, 2009.

[27] R. S. G. Jones, “Entorhinal-hippocampal connections: a speculative view of their function,” *Trends in Neurosciences*. 1993.

[28] E. Save and F. Sargolini, “Disentangling the role of the MEC and LEC in the processing of spatial and non-spatial information: Contribution of lesion studies,” *Frontiers in Systems Neuroscience*. 2017.

[29] A. Tsao *et al.*, “Integrating time from experience in the lateral entorhinal cortex,” *Nature*, 2018.

[30] E. Kropff, J. E. Carmichael, M. B. Moser, and E. I. Moser, “Speed cells in the medial entorhinal cortex,” *Nature*, 2015.

[31] Y. Aoki, H. Igata, Y. Ikegaya, and T. Sasaki, “The Integration of Goal-Directed Signals onto Spatial Maps of Hippocampal Place Cells,” *Cell Rep.*, 2019.

[32] Я. Б. Казанович and Y. B. Kazanovich, “How Animals Find Their Way in Space Experiments and Modeling,” *Математическая биология и биоинформатика*, vol. 10, no. 1, pp. 88–115, Mar. 2015.

[33] M. Kim and E. A. Maguire, “Encoding of 3D head direction information in the human brain,” *Hippocampus*, 2019.

[34] H. J. I. Page, J. J. Wilson, and K. J. Jeffery, “A dual-axis rotation rule for updating the

head direction cell reference frame during movement in three dimensions," *J. Neurophysiol.*, 2018.

[35] C. Lever, S. Burton, A. Jeewajee, J. O'Keefe, and N. Burgess, "Boundary vector cells in the subiculum of the hippocampal formation," *J. Neurosci.*, 2009.

[36] W. Mau, D. W. Sullivan, N. R. Kinsky, M. E. Hasselmo, M. W. Howard, and H. Eichenbaum, "The Same Hippocampal CA1 Population Simultaneously Codes Temporal Information over Multiple Timescales," *Curr. Biol.*, 2018.

[37] D. Marr, "Simple memory: a theory for archicortex.," *Philos. Trans. R. Soc. Lond. B. Biol. Sci.*, 1971.

[38] R. C. O'Reilly and J. L. McClelland, "Hippocampal conjunctive encoding, storage, and recall: Avoiding a trade-off," *Hippocampus*, vol. 4, no. 6, pp. 661–682, Dec. 1994.

[39] A. Kaplan and M. Haenlein, "Siri, Siri, in my hand: Who's the fairest in the land? On the interpretations, illustrations, and implications of artificial intelligence," *Business Horizons*. 2019.

[40] M. Z. Altan, H. Gardner, and M. Z. Altan, "Intelligence Reframed: Multiple Intelligences for the 21st Century," *TESOL Q.*, 2001.

[41] A. C. Müller and S. Guido, *Introduction to Machine Learning with Python: A Guide for Data Scientists*. 2017.

[42] T. Bekolay *et al.*, "Nengo: A Python tool for building large-scale functional brain models," *Front. Neuroinform.*, 2014.

[43] B. Morcos, T. C. Stewart, C. Eliasmith, and N. Kapre, "Implementing NEF Neural Networks on Embedded FPGAs," in *Proceedings - 2018 International Conference on Field-Programmable Technology, FPT 2018*, 2018.

[44] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: Part I," *IEEE Robot. Autom. Mag.*, 2006.

[45] D. Kumaran, D. Hassabis, and J. L. McClelland, "What Learning Systems do Intelligent Agents Need? Complementary Learning Systems Theory Updated," *Trends in Cognitive Sciences*. 2016.

[46] R. C. O'Reilly, R. Bhattacharyya, M. D. Howard, and N. Ketz, "Complementary learning systems," *Cogn. Sci.*, 2014.

[47] G. Kowadlo, A. Ahmed, and D. Rawlinson, "AHA! an 'Artificial Hippocampal Algorithm' for Episodic Machine Learning." 2019.

[48] D. Ramani, "A Short Survey On Memory Based Reinforcement Learning." 2019.

[49] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities.," *Proc. Natl. Acad. Sci. U. S. A.*, 1982.

[50] A. Treves and E. T. Rolls, "Computational constraints suggest the need for two distinct input systems to the hippocampal CA3 network," *Hippocampus*, 1992.

[51] Q. Chen and H. Mo, "A Brain-Inspired Goal-Oriented Robot Navigation System," *Appl. Sci.*, vol. 9, no. 22, p. 4869, Nov. 2019.

[52] M. J. Milford, G. F. Wyeth, and D. Prasser, "RatSLAM: A hippocampal model for

simultaneous localization and mapping," in *Proceedings - IEEE International Conference on Robotics and Automation*, 2004.

[53] M. Salman and M. J. Pearson, "Whisker-ratSLAM applied to 6D object identification and spatial localisation," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2018.

[54] H. Tulleken, "Hex Grid Geometry for Game Developers," no. June, pp. 1–35, 2015.

[55] L. M. Giocomo, M. B. Moser, and E. I. Moser, "Computational models of grid cells," *Neuron*. 2011.

[56] P. K. Pilly and S. Grossberg, "How does the modular organization of entorhinal grid cells develop?," *Front. Hum. Neurosci.*, 2014.

[57] R. S.Sutton and A. G.Barto, *Reinforcement Learning: An Introduction*. 2015.

[58] K. Duncan, B. B. Doll, N. D. Daw, and D. Shohamy, "More Than the Sum of Its Parts: A Role for the Hippocampus in Configural Reinforcement Learning," *Neuron*, 2018.

[59] P. Dayan and Y. Niv, "Reinforcement learning: The Good, The Bad and The Ugly," *Current Opinion in Neurobiology*. 2008.

[60] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-Learning," in *30th AAAI Conference on Artificial Intelligence, AAAI 2016*, 2016.

[61] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Mach. Learn.*, 1992.

[62] H. Jiang, R. Gui, Z. Chen, L. Wu, J. Dang, and J. Zhou, "An Improved Sarsa Reinforcement Learning Algorithm for Wireless Communication Systems," *IEEE Access*, 2019.

[63] J. M. Olson, K. Tongprasearth, and D. A. Nitz, "Subiculum neurons map the current axis of travel," *Nat. Neurosci.*, 2017.

[64] T. Hafting, M. Fyhn, S. Molden, M. B. Moser, and E. I. Moser, "Microstructure of a spatial map in the entorhinal cortex," *Nature*, 2005.

[65] M. F. Goodchild and A. J. Kimerling, *Discrete Global Grids: A Web Book*. 2002.

[66] M. G. Paulin, "Neural Engineering: Computation, Representation and Dynamics in Neurobiological Systems," *Neural Networks*, 2004.

[67] T. Kitamura *et al.*, "Island cells control temporal association memory," *Science (80-.).*, 2014.

[68] R. K. Naumann, P. Preston-Ferrer, M. Brecht, and A. Burgalossi, "Structural modularity and grid activity in the medial entorhinal cortex," *Journal of Neurophysiology*. 2018.

[69] A. Bakker, C. B. Kirwan, M. Miller, and C. E. L. Stark, "Pattern separation in the human hippocampal CA3 and dentate gyrus," *Science (80-.).*, 2008.

[70] C. E. Myers and H. E. Scharfman, "Pattern separation in the dentate gyrus: A role for the CA3 backprojection," *Hippocampus*, 2011.

[71] M. A. Yassa and C. E. L. Stark, "Pattern separation in the hippocampus," *Trends in Neurosciences*. 2011.

[72] P. E. Danielsson, “Euclidean distance mapping,” *Comput. Graph. Image Process.*, 1980.

[73] P. E.Black, “Manhattan Distance,” *Dictionary of Algorithms and Data Structures*. [Online]. Available: https://xlinux.nist.gov/dads/HTML/manhattanDistance.html. [Accessed: 14-Sep-2020].

[74] K. L. Elmore and M. B. Richman, “Euclidean distance as a similarity metric for principal component analysis,” *Mon. Weather Rev.*, 2001.

[75] M. Kokare, B. N. Chatterji, and P. K. Biswas, “Comparison of similarity metrics for texture image retrieval,” in *IEEE Region 10 Annual International Conference, Proceedings/TENCON*, 2003.

[76] D. A.Sousa, *How the Brain Works 5th Edition*. 2017.

[77] J. M. J. Murre and J. Dros, “Replication and analysis of Ebbinghaus’ forgetting curve,” *PLoS One*, 2015.

[78] P. Xia, L. Zhang, and F. Li, “Learning similarity with cosine similarity ensemble,” *Inf. Sci. (Ny).*, 2015.

[79] R. W. Hamming, “Error Detecting and Error Correcting Codes,” *Bell Syst. Tech. J.*, 1950.

[80] A. Konar, I. G. Chakraborty, S. J. Singh, L. C. Jain, and A. K. Nagar, “A deterministic improved q-learning for path planning of a mobile robot,” *IEEE Trans. Syst. Man, Cybern. Part ASystems Humans*, 2013.

[81] A. A. Hagberg, D. A. Schult, and P. J. Swart, “Exploring network structure, dynamics, and function using NetworkX,” in *7th Python in Science Conference (SciPy 2008)*, 2008.

[82] O. Michel, “Webots robot simulator,” *https://www.cyberbotics.com*. 2015.

[83] Cyberbotics, “K-Team’s Khepera IV,” *Webots User Guide*. [Online]. Available: https://www.cyberbotics.com/doc/guide/khepera4. [Accessed: 28-Apr-2020].

[84] Cyberbotics, “Compass.” [Online]. Available: https://cyberbotics.com/doc/reference/compass.

[85] W. Gao, L. Yang, X. Zhang, and H. Liu, “An improved Sobel edge detection,” in *Proceedings - 2010 3rd IEEE International Conference on Computer Science and Information Technology, ICCSIT 2010*, 2010.

[86] Scipy, “Cosine Method.” [Online]. Available: https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.distance.cosine.html .

[87] C. Barry *et al.*, “The boundary vector cell model of place cell firing and spatial memory,” *Reviews in the Neurosciences*. 2006.

[88] Pyhton, “Python Random.” [Online]. Available: https://docs.python.org/3/library/unittest.html.

[89] Python, “Unittest Framework.” [Online]. Available: https://docs.python.org/3/library/unittest.html.

[90] Scipy, "Scipy Euclidean Distance." [Online]. Available: https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.distance.euclidean.html.

[91] Scipy, "Scipy Manhattan Distance." [Online]. Available: https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.distance.cityblock.html.

APPENDIX

A. UNIT TESTS FOR THE IMPLEMENTED MODELS

This section contains the created tests for the implemented models mentioned above. These programming tests were conducted with a small dataset of robot values obtained from the Webots robot simulator. This section contains the unit tests made for the Entorhinal Cortex and the Hippocampus models.

A.1 TESTING OF THE ENTORHINAL CORTEX MODEL

The Entorhinal Cortex model is divided into two parts: (1) the Grid Cell part which contains the necessary calculations for the creation of simulated Grid Cells according to the previously mentioned model, and (2) the encoding part responsible for translating sensory to a ternary configuration.

The most part of the created tests to the Grid Cell model were made with the Webots simulator. However, before the incorporation with the Webots simulator, a smaller test unit was created. The direction of movement was simulated using a pseudorandom function provided by the Python programming language [88]. The angles started with $\frac{\pi}{3}$, $\frac{5\pi}{3}$ and the moving speed was set to $\frac{\pi}{12}$. For plotting reasons, when the simulations values exceeded the environmental grid values, a new direction was calculated.

For the encoding part, two tests were made, one for the cyclic and non-cyclic. The test consists in selecting a number, in this case the chosen number was 23. Such results were checked with Table 3.1 and Table 3.2. The structure of the unit test is illustrated in Figure A.1

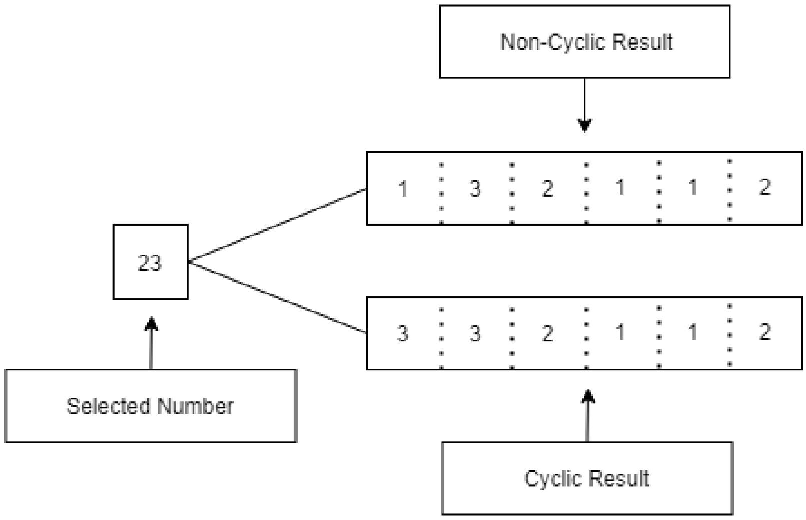


Figure A.1: Unit Test and Expected Result of the Encoding Method

A.2 TESTING OF THE HIPPOCAMPUS MODEL

The main tests were made with the Webots simulator. However, because of the complexity of such model, a smaller dataset was used. The dataset elements that supported the tests for the Hippocampus model were multiple elements captured directly from the robot simulator. With such data, it was possible to test the integrity of the model. However, other tests were made to the individual functions of system. These smaller tests ensure the correctness of the implemented mechanisms. The comparison between the test result and the system's result is made with the help of a Python library called unittest [89]. This library allows the creation of different tests scenarios for all system or to some system functions. Figure A.2 shows the implementation of a unit test to verify the integrity of the encoded method.

```python
def test_encoding_non_cyclic_data(self):
    """
        Checks the encoding method on a non-cyclic value.

        :return: Must be equal to [1, 3, 2, 1, 1, 2]
    """
    number = 23
    cyclic = False
    obtained_result = ec.encoding_method(number, cyclic)
    expected_result = [1, 3, 2, 1, 1, 2]
    self.assertEqual(obtained_result, expected_result, "Encoding must be the same")

def test_encoding_cyclic_data(self):
    """
         Checks the encoding method on a cyclic value.

        :return: Must be equal to [3, 3, 2, 1, 1, 2]
    """
    number = 23
    cyclic = True
    obtained_result = ec.encoding_method(number, cyclic)
    expected_result = [3, 3, 2, 1, 1, 2]
    self.assertEqual(obtained_result, expected_result, "Encoding must be the same")
```

Figure A.2: Unit Test for the Gray Scale Method

A.2.1 Testing of the Dentate Gyrus

In the DG section, there are not so many methods because, as mentioned before, the DG area only acts as a system disambiguation method. The DG class has two major calculations. The Cosine Similarity calculation, that was implemented with the help of a Python library called Scipy, and the Hamming distance calculation [86]. However, the Hamming distance function does not depend on the library and for that reason a unit test was created. The Figure A.3 shows the result of the implemented function should be the number of elements that are not equal in both arrays.

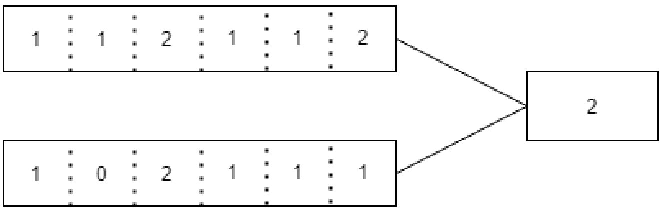


Figure A.3: Unit Test for the Hamming Distance

A.2.2 TESTING OF THE CA3

As mentioned before, the CA3 area finds similarities between the stored memory elements and the current data by using their recurrent connections. To achieve such behaviour, it uses the Euclidean [90] and the Manhattan [91] distances, calculated using the Scipy Spatial Distances library. Meanwhile, there are some used supporting functions. For such supporting functions, two tests were created. One verifies if the sort mechanism orders the two least distance values obtained from the CA3 calculations. The second test was created in order to verify the integrity of the complementary result system.

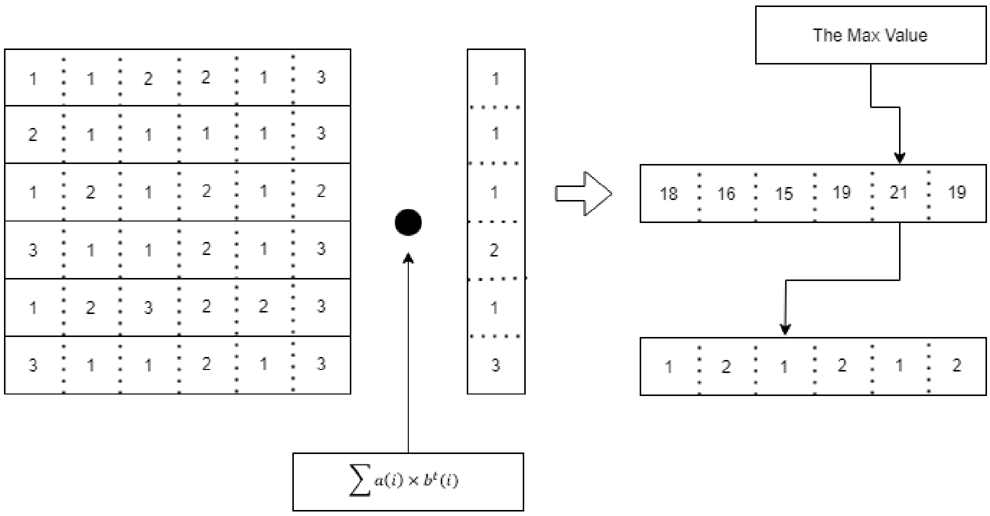


Figure A.4: Unit Test of the Result Complementary System

Figure A.5 illustrates the mechanism for sorting the elements. A list of elements was used, and its purpose is to extract the two smaller elements and their memory elements positions.

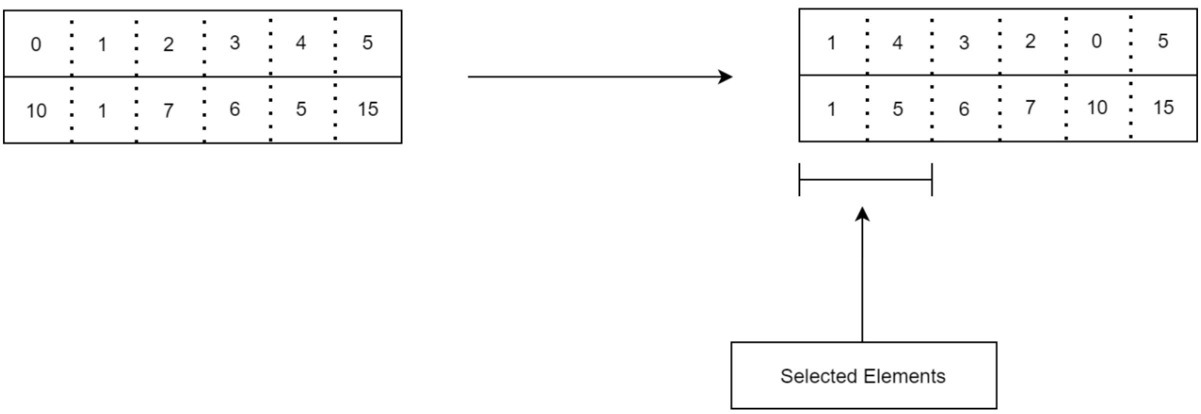
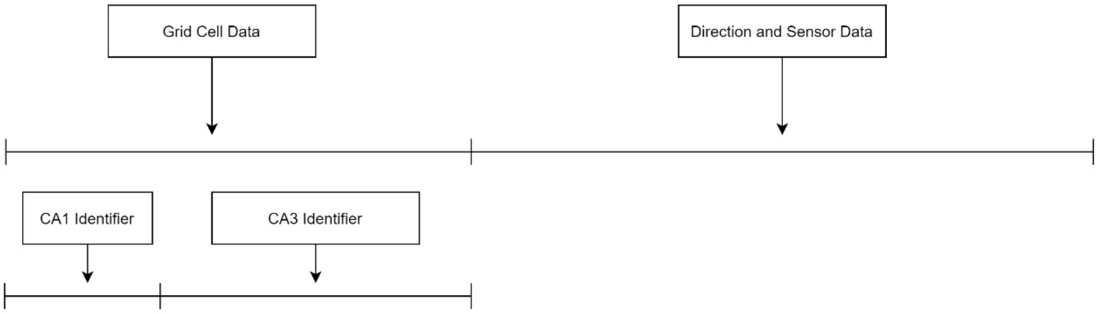


Figure A.5: Unit Test for Sort Elements Method

A.3 STRUCTURAL TEST

To verify the correctness of the entire system, a set of data was directly extracted from the robot simulator program was used. Figure A.6 represents a data stream extracted for the specific test. Each data line contains information of Grid Cells, the direction of movement and the sensors data captured from the Khepera IV robot.



A	1	1	3	1	1	3	1	1	3	3	1	1	3	3	1	2	2	2	2	3	1
B	1	1	3	1	1	3	1	1	3	3	1	1	3	3	1	2	1	1	1	3	3
C	2	1	2	3	1	1	3	1	1	3	1	1	3	2	1	1	2	3	3	2	2
D	2	1	2	3	1	1	3	1	1	3	1	1	3	2	1	3	3	3	1	2	3
E	2	1	2	3	1	1	3	1	1	3	1	1	3	2	2	1	1	1	1	2	2
F	2	1	2	3	1	1	3	1	1	3	1	1	1	2	2	1	3	3	3	3	1
G	1	1	3	3	1	1	3	1	1	3	1	1	1	2	2	3	1	1	1	2	2
H	1	1	3	3	1	1	3	1	1	3	1	1	1	2	2	2	1	3	2	2	3
I	1	1	1	2	2	2	2	3	3	3	1	1	1	2	3	2	1	3	3	1	2
J	1	1	1	2	2	2	2	3	3	3	1	1	1	2	3	3	1	1	1	2	3
K	1	1	3	1	2	3	2	3	3	3	1	1	1	1	1	3	3	3	2	2	3
L	1	1	3	1	1	3	1	1	3	3	1	1	3	3	1	3	3	1	1	1	2
M	1	1	1	2	2	2	2	3	3	3	1	1	1	2	3	3	3	1	2	3	1
N	1	1	1	2	2	2	2	3	3	3	1	1	1	1	3	1	1	1	3	3	1
O	1	1	1	2	2	2	2	3	3	3	1	1	1	1	3	1	3	2	2	2	2
P	2	1	2	3	1	1	3	1	1	3	1	1	3	2	1	1	3	3	1	1	1

Figure A.6: Chosen test dataset and the expected result for the Encoding Method

For illustration purposes, each data element is identified by a letter, making it easier to find a specific element inside the Figure A.6. The second phase is the identification of similar events by using the distance calculation from the CA3. In the final phase, the CA3 sends data to the

CA1 area for Place Cell recognition and to the Subiculum to create the self-awareness map. Figure A.8 illustrates how CA3 elements are stored in memory and the obtained of the previous letters identified by a specific letter.

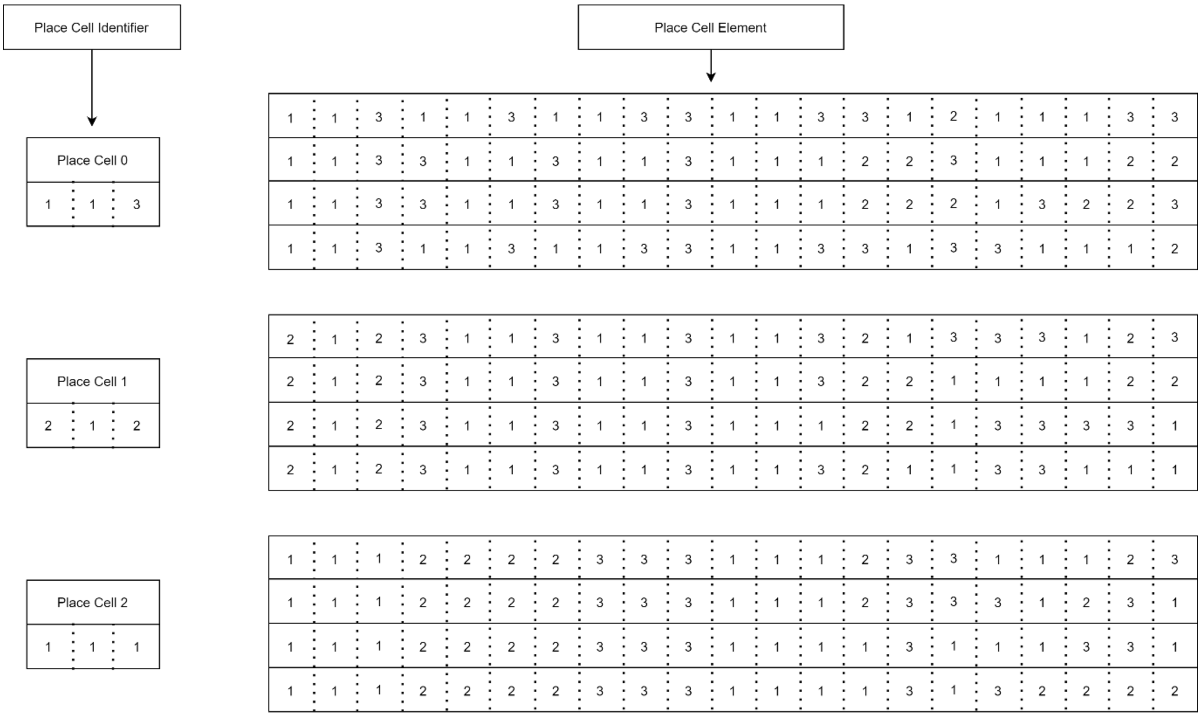


Figure A.7: CA1 Expected Result

Figure A.7 represents the expected result for the CA1 area. Using the top layer as identification selector, each element is inserted in a specific spot. Such spot works as a simulated Place Cell.

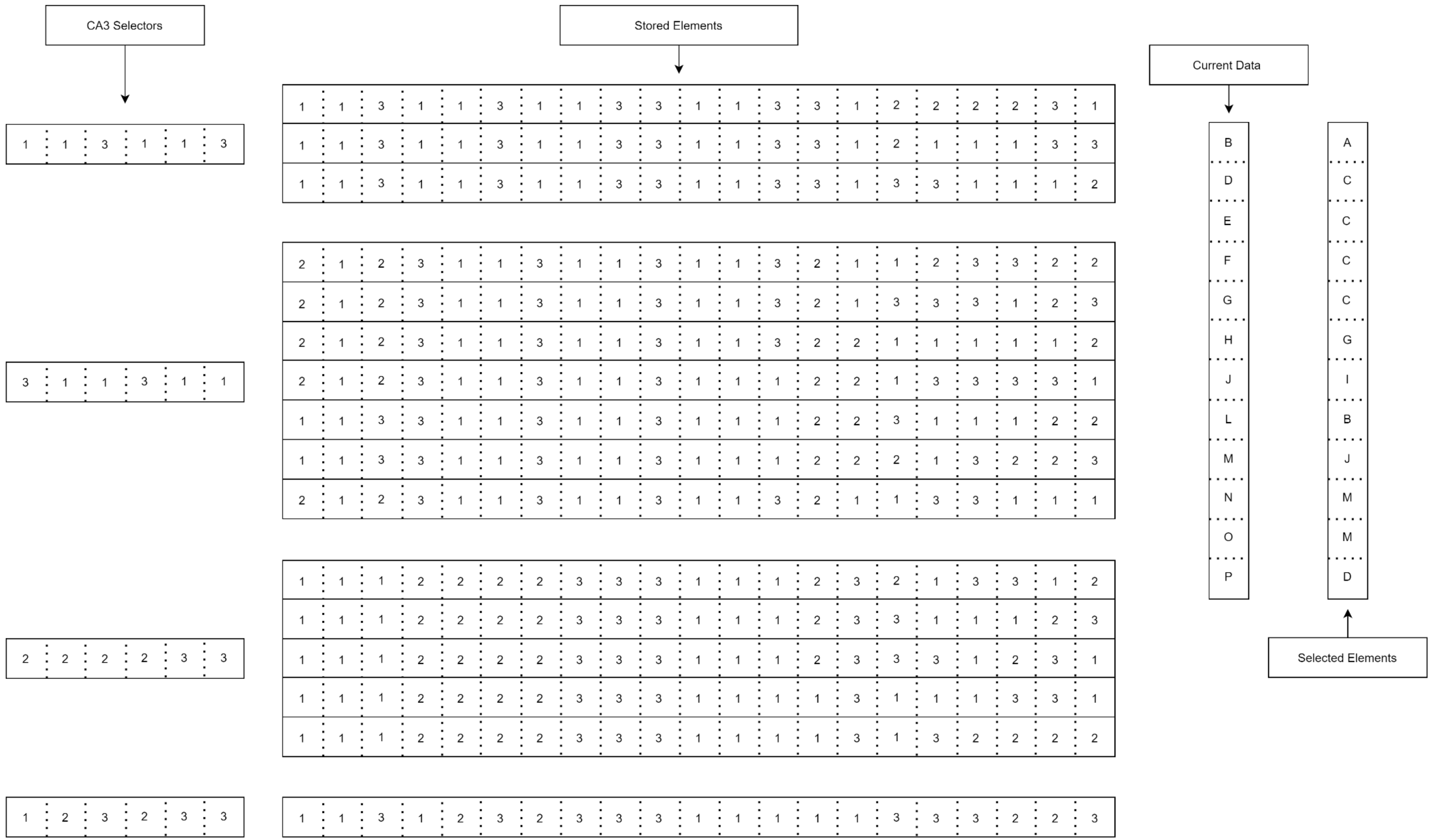


Figure A.8: CA3 and Memory Expected Result