Spike-based computational models of bio-inspired memories in the hippocampal CA3 region on SpiNNaker

Daniel Casanueva-Morato, Alvaro Ayuso-Martinez, Juan P. Dominguez-Morales,
Angel Jimenez-Fernandez and Gabriel Jimenez-Moreno
Robotics and Technology of Computers Lab.
Universidad de Sevilla
Sevilla, España
dcasanueva@us.es

Abstract—The human brain is the most powerful and efficient machine in existence today, surpassing in many ways the capabilities of modern computers. Currently, lines of research in neuromorphic engineering are trying to develop hardware that mimics the functioning of the brain to acquire these superior capabilities. One of the areas still under development is the design of bio-inspired memories, where the hippocampus plays an important role. This region of the brain acts as a short-term memory with the ability to store associations of information from different sensory streams in the brain and recall them later. This is possible thanks to the recurrent collateral network architecture that constitutes CA3, the main sub-region of the hippocampus. In this work, we developed two spike-based computational models of fully functional hippocampal bio-inspired memories for the storage and recall of complex patterns implemented with spiking neural networks on the SpiNNaker hardware platform. These models present different levels of biological abstraction, with the first model having a constant oscillatory activity closer to the biological model, and the second one having an energy-efficient regulated activity, which, although it is still bio-inspired, opts for a more functional approach. Different experiments were performed for each of the models, in order to test their learning/recalling capabilities. A comprehensive comparison between the functionality and the biological plausibility of the presented models was carried out, showing their strengths and weaknesses. The two models, which are publicly available for researchers, could pave the way for future spike-based implementations and applications.

Index Terms—Hippocampus model, CA3, Neuromorphic engineering, spiking neural networks, SpiNNaker, spike-based memory

I. INTRODUCTION

Neuromorphic engineering is a concept that was presented by Carver Mead in [1], and it is a field that focuses on studying, designing and implementing hardware and software with the aim of mimicking the basic principles of biological nervous systems. Its main inspiration comes from studying and replicating how the brain efficiently solves complex problems [2]. Based on this biological approach, the information in neuromorphic systems is represented with action potentials (i.e., asynchronous electric pulses, also called spikes) generated by artificial neurons. Spikes are transmitted between different processing layers, leading up to a higher-level functionality.

This approach has a clear advantage both in terms of power consumption and real-time capabilities when compared to traditional methods [3] [4].

In order to process the aforementioned spiking information, a bio-inspired computational approach is also needed. A specific type of biologically-plausible neural networks called Spiking Neural Networks (SNNs) are commonly used for this purpose. These are able to process asynchronous spikes instead of floating point or integer numbers, as traditional artificial neural networks would do. SNNs consists of bio-inspired neuron models, which are interconnected by means of synapses, and mimic the way in which the senses and the brain process the information in living organisms [5]. In the last years, SNNs have been used for many different tasks, such as speech recognition [6], [7], image classification [8], [9], sensory fusion [8], [10], motor control [11], [12] and bio-inspired locomotion [13]–[15], among many others, proving that they could be useful in a wide variety of fields and allowing more efficient and lower power-consumption devices to perform in the same manner as their traditional counterparts.

Among all the different regions within the brain, one of the most interesting ones in terms of its biological behavior is the hippocampus. This component functions primarily as a short-term memory, storing rapid (on-the-fly) and unstructured information coming from the different sensory streams of the cerebral cortex. The hippocampal formation is composed of three main parts: Dentate Gyrus (DG), the Hippocampus proper (Cornu Ammonis (CA)) and the Subiculum. It is in CA3, a subregion of CA, where the recurrent collateral network structure capable of storing information is located [16]. Making use of the mechanisms underlying this sector of the brain to create an efficient and functional bio-inspired memory with SNN technology could pave the way for future low-power and low-latency applications.

Previous works that can be found in the literature already proposed different bio-inspired hippocampus memory models. In [17], a hybrid memory architecture is proposed, which combines the structure of CA3 as an oscillating network to repeat sequences of input patterns and the memory structure

of the neocortex to store them by modifying its synaptic weights. However, it is only capable of storing simple pattern sequences, and features a neuron model whose oscillation rate is controlled by an external signal from outside the network. In [18], a model of the hippocampus consisting of DG and CA3 is proposed. However, it is not purely spike-based, since the equations that govern CA3 are modeled in digital logic and DG is developed as a traditional Artificial Neural Network (ANN). Other works regarding this topic can be found in the literature, but some are approached from a purely theoretical point of view, such as [19]; others only perform proofs of concept, such as [20], where a spatial navigation mechanism is evaluated using a static CA3 model with previously stored information in which no learning is applied.

The lack of fully functional bio-inspired spike-based hippocampal memory models that could be used for real-case scenarios including robotics or many other applications was the main motivation for this work. Therefore, two different bio-inspired memory models are proposed: the first model has a closer biological approach, whereas the second model follows a more functional approach. The former is characterised by a constant oscillatory activity in the CA3 attractor network, while the latter adds additional mechanisms on top of the first model to precisely control the network activity. These models have been implemented on the SpiNNaker hardware platform and tested in a series of experiments aimed at demonstrating their performance. The results of the experiments will be used as the basis for analyzing the advantages and disadvantages of moving towards or away from biology in search of functionality.

The main contributions of this work include the following:

- Two fully-functional spike-based bio-inspired hippocampal memory models are proposed. These are capable of storing and retrieving complex patterns with certain limitations.
- The proposed models were not simply simulated in software, but implemented and emulated on the SpiNNaker hardware platform.
- The source code of the proposed models is publicly available, together with all the necessary details regarding the SNN architectures.

II. BIOLOGICAL MODEL OF THE HIPPOCAMPUS AND ITS FUNCTION AS SHORT-TERM MEMORY

The hippocampus belongs to the limbic system, the region of the brain that controls emotional responses and is involved in smell, appetite, eating habits, sleep and certain areas of the memory [21]. Specifically, the hippocampus or hippocampal system is a structure inside the brain that consists of the 3 following layers [22] [23], as can be seen in Fig. 1:

- Dentate Gyrus (DG): it is the input area for the information coming from the Entorhinal Cortex (EC). It plays an important role in the mechanism of pattern separation of the input information; specifically, it is responsible for dispersing the content of the information to achieve a greater degree of orthogonalization, i.e., converting the

input information into a representation as different as possible from any other input information. The purpose of this mechanism is to improve both storage capacity and memory access.
- Hippocampus (CA): it consists of Pyramidal Cells (PCs), and is divided into CA1, CA2, CA3 and CA4, with CA1 and CA3 being the most relevant in terms of functionality. This region is where the input information is stored and the stored information is recalled and recoded, when requested from the outside.
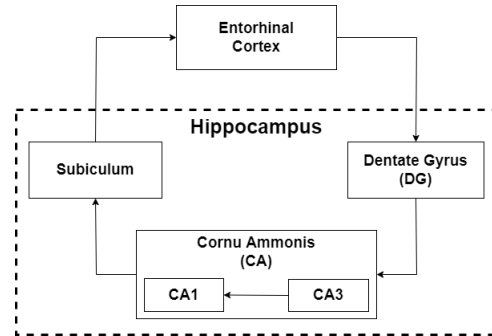- Subiculum: it receives the output from the hippocampus and redirects it back to the EC.

Fig. 1. Diagram of the different layers of the hippocampus and the main connecting pathways between them.

The EC (Brodmann's area 28) acts as the main information input and output pathway of the hippocampal system. It has as many input (forward) as output (backward) projections. Through this cortex, the hippocampus receives information from all the different sensory information streams in the brain (including spatial) coming from the neocortex and sends the processed information back to these sensory stream units.

CA3 acts as an autoassociative or attractor memory due to its recurrent collateral network structure, as can be seen in Fig. 1. This region also contains recurrent inhibitory interneurons that regulate the global activity during periods of network oscillation. By receiving all the information from the brain's sensory streams and having the ability to store them in an associated and unstructured way, the hippocampus acts as a short-term memory involved in episodic memory (memory of specific autobiographical events of the past). Therefore, it must be able to store this information (CA3) in a distinct way (separating memories corresponding to different moments thanks to DG) and retrieve this information later in order to use it (via CA1 and the subiculum) [16].

III. MATERIALS AND METHODS

A. Spiking Neural Networks

In order to implement bio-inspired memory models based on the hippocampus, third-generation of neural networks, i.e., SNNs, were used. This kind of networks consists of interconnected neuron models that are not only inspired by biology, but also attempt to mimic their biological counterparts, in order
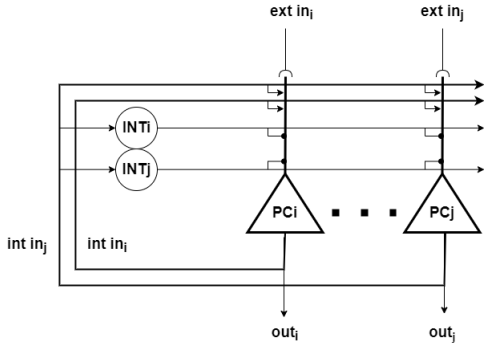
Fig. 2. Internal structure of CA3. It consists of a recurrent collateral network of PCs with external inputs from the DG (ext) and internal inputs from the collateral connections (int), as well as collateral inhibitory inputs from a population of interneurons (INT) that regulate the rate of activity of the network. All the connections are excitatory (lines ending with an arrowhead) except for the projections between the interneurons back to the input of the PCs, which are inhibitory (lines ending with a circle).

to incorporate the neurocomputational capabilities found in nature [5].

As was introduced in section I, SNNs present an asynchronous event-based functioning and communication thanks to the use of spikes. With the spiking activity, they incorporate the concept of space and time into the neural networks through connectivity and plasticity. It is possible to interpret SNN models, i.e., to extract the set of rules that govern them and to understand the learning mechanism [24]. These event-based networks only have to compute the information when an event occurs (a spike is generated and/or received). This aspect makes SNNs very efficient from a computational point of view. Moreover, SNNs present a distributed learning process thanks to the use of Spike-timing-dependent plasticity (STDP) [25], which obtains the necessary information from the electrical impulses encoded between local neurons to define the weight of the synapses, i.e., to regulate the learning of the network [5].

SNNs present key features for being implemented in hardware compared to traditional ANNs, allowing for higher efficiency and lower power consumption: multiplications are replaced with adders and shifts, the information that is transmitted between neurons is only 1-bit in size instead of integer or floating point numbers, etc. [24].

Different hardware platforms particularly designed for implementing and simulating SNNs can be found in the literature. Some of the most well-known ones are SpiNNaker [26], Loihi [27] and TrueNorth [28]. In this work, we used SpiNNaker as the hardware platform in which the different SNN models presented were implemented and emulated.

B. SpiNNaker

SpiNNaker [26] is a massively-parallel multi-core computing system, which was designed for being able to model very large SNNs in real time. Each SpiNNaker chip consists of 18 general-purpose ARM968 cores, running at 200 MHz, which communicate the information by means of packets carried by

a custom interconnect fabric [29]. In this work, a SpiNN-5 machine was used. SpiNN-5 is a 48-node circuit board and, thus, has 864 ARM processor cores, which are commonly deployed as 768 application cores, 48 Monitor Processors and 48 spare cores. A 100 Mbps Ethernet connection is used as an I/O interface and for sending scripts and commands to the board. A custom software package called sPyNNaker [30] allows running PyNN [31] simulations directly on the SpiNNaker board. This makes the platform very straightforward to work with, since all the codes regarding the design and implementation of SNNs can be done using high-level functions described in Python programming language.

IV. HIPPOCAMPUS-BASED COMPUTATIONAL MODELS OF MEMORY

This paper presents two bio-inspired computational memory models of the hippocampus with different degrees of closeness to the theoretical bases. Before describing all the implementation details of these models, it is necessary to define the neuron models, synapses and learning rules used first.

Starting with neurons, many different models can be found in the literature, each of them with specific characteristics depending on the level of abstraction. Among them, the most widely-used in the field of SNNs, and the one that has been used for the proposed model implementations is the standard Leaky Integrate-and-Fire (LIF). This model describes the behavior of a neuron as an RC electrical circuit whose potential value is either increased or decreased with the arrival of input spikes and, after reaching a threshold value, generates output spikes. For both of the implemented spike-based memory models, the parameters of the LIF neurons used are: c_m: 0.27 nF, τ_m: 10.0 ms, τ_{synExc}: 0.3 ms, τ_{synInh}: 0.3 ms, v_{reset}: -60.0 mV, v_{rest}: -60.0 mV, v_{thresh}: -55.0 mV, where c_m is the capacitance, τ_m is the time-constant of the RC circuit, τ_{synExc} and τ_{synInh} are the excitatory and inhibitory input current decay time-constants, respectively, v_{reset}, v_{rest} and v_{thresh} are the membrane potentials at which the neuron is set immediately after generating a spike, during the inactivity period and the threshold at which the neuron spikes, respectively. The value of the τ_{refrac} parameter, which refers to the refractory period of the neuron (the time that the neuron is disabled after generating a spike and stops integrating any input spike) is set different for the two SNN models proposed.

Synapses can be modelled as a weighted edge in a graph connecting a source (pre-synaptic) neuron to a target (post-synaptic) neuron. The learning mechanism of SNNs focuses on the plasticity of the synapses, i.e., the ability to create, remove or modify the weight of the synapses. This weight represents an increase or a decrease in the action potential that will be applied to the postsynaptic neuron when a spike is received. Therefore, a modification of the weight would pose an influential change.

Among all the different learning mechanisms where the concept of synapse plasticity is considered, the most widely-known and the one that was used in this work is STDP.

It is based on a Hebbian learning mechanism in which the weights of synapses are modified in proportion to the temporal correlation between pre- and post-synaptic neuronal activity [32]. The weight of a synapse will increase whenever a presynaptic spike is received before the postsynaptic spike, and will decrease otherwise, by an amount proportional to the time difference between the two spikes (the closer in time, the greater). STDP is implemented in SpiNNaker, although it presents some particularities: the evaluation of the weight change (sum of cumulative increments and decrements) in a synapse is produced only on presynaptic spikes [33].

Based on the materials and methods described, two different CA3 models were proposed and implemented: an oscillatory CA3 model (Section IV-A), and a regulated CA3 model (Section IV-B).

A. CA3 memory model with oscillatory activity

The first of the two models implemented (see Fig. 3) is the closest to the biological behavior, as described in the literature, i.e., an attractor network that maintains an internal state and is constantly oscillating. The design consists of a layer of DG neurons that act as input to the system and a layer of PCs that form CA3. DG has one-to-one excitatory connections with PC neurons, and PC neurons are interconnected between them by means of one-to-all recurrent collateral connections, both excitatory and inhibitory. The excitatory collateral synapses between PC neurons are where STDP is applied, and they are responsible for the storage of patterns in the network. The rest of the synapses are static.
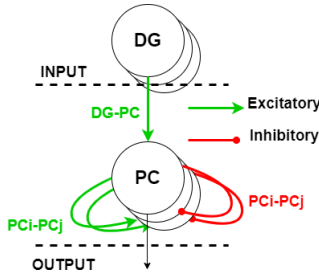

Fig. 3. Diagram of the DG-CA3 network with oscillatory activity in the recurrent collateral connections of CA3. Excitatory synapses are shown in green and inhibitory synapses in red. The excitatory synapses where STDP is applied are $PC_i - PC_j$.

The static synapses of the network are set with a delay of 1 ms. Regarding the weights, the recurrent inhibitory connections ($PC_i - PC_j$) are set with a weight of 1.5 nA, and the connections between DG and PC neurons are set to a value that is high enough for a presynaptic spike to always generate a postsynaptic spike. As for the STDP learning rule, the parameters used were: τ_plus = 3.0 ms, τ_minus = 2.0 ms, A_plus = 6.0, A_minus = 3.0, maximum weight = 12.0 nA, minimum weight = 0.0 nA, initial weight = 0.0 nA and delay = 1.0 ms, where τ_plus and τ_minus are the decay time-constants that control the amount of weight increase or decrease, respectively, and A_plus and A_minus define the

maximum weight to respectively add during potentiation or subtract during depression.

This network has two main phases: a learning phase and a recall phase. In the former, the patterns to be learned are introduced to the network. The STDP rule changes the weights of the synapses and, thus, the internal state of the attractor. In the second phase, a part of the pattern to be recalled (cue) is introduced as input. After a few oscillation cycles, the network returns the rest of the pattern. Once the first cue arrives at the network, it starts oscillating and it is the passing of the cue that makes it oscillate from one state to another, with the help of the lateral inhibitions, where the state is the pattern to be remembered. For this to happen, the neurons need to be configured with a refractory period of 0, so that they will be constantly oscillating between states.

B. CA3 memory model with regulated periods of activity

The second model implemented (see Fig. 4) is an extension of the previous model, where an activity inhibition mechanism is added, changing the refractory period from 0 to 2 ms and using stronger lateral inhibitions that will depend on the number of PC neurons. These changes aim to precisely regulate and control the internal activity of the CA3 recurrent collateral network, reducing power consumption and achieving greater stability.
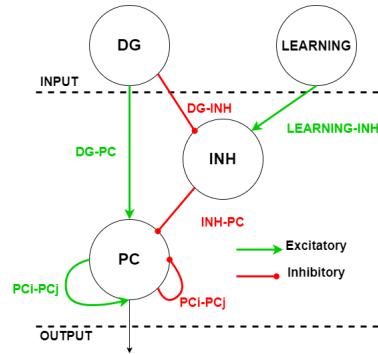

Fig. 4. Diagram of the DG-CA3 network with regulated activity to avoid constant oscillation. Excitatory synapses are shown in green and inhibitory synapses in red. The excitatory synapses where STDP is applied are $PC_i - PC_j$. It has an additional mechanism with inhibitory interneurons (INH) to regulate the activity in the learning phase.

The inhibition mechanism consists of a layer of inhibitory interneurons (INH) of the same size as the DG and the PC layer, and receives two inputs: an excitatory signal indicating whether the network is in the learning phase (LEARNING), and an inhibitory input with one-to-one connectivity from the DG (DG-INH). The output of the network is sent to PC neurons via inhibitory one-to-one synapses (INH-PC). The neuron models and static synapses are identical to those used for the DG-PC connections. This inhibitory mechanism mimics the inhibitory interneurons that are present in some biological models of the hippocampus in order to regulate the oscillation rate of the network only while learning, but not during the recall phase. This approach is also bio-inspired,

but it is not as close to biology as the previous model, mainly due to the introduction of the LEARNING signal.

The modifications performed in this model with respect to the first model allow having a precise event-based activity network, rather than a constant oscillation, with the consequent advantages and disadvantages that are detailed and discussed in section V. It also has a learning phase and a recall phase in which the STDP remains static. This means that the weights obtained after the learning phase are used in the same network but with static synapses to perform the recall without the problem of forgetting data due to it.

V. EXPERIMENTATION AND RESULTS

In order to test the behavior of the models implemented in SpiNNaker, different experiments were performed. The first set of experiments were focused on verifying the performance of the proposed models using completely orthogonal patterns (patterns that are completely different from each other and have nothing in common) as input, which is the minimum requirement for the memory to be functional. Then, the second set of experiments were focused on testing the models when using non-orthogonal patterns (patterns that have some parts in common) as input. For all the experiments that were performed, networks with 15 PC neurons were used.

A. CA3 memory model with oscillatory activity

In the first set of experiments, the oscillatory network was able to learn and correctly recall different combinations of 2-4 fully orthogonal patterns. In the top plot of Fig. 5, the results for the case with 3 orthogonal patterns can be observed. In the first 6 milliseconds (milliseconds 1-6) the pattern 0-1-2-3-4 was used as input to the network, simulating the arrival of those 5 spikes coming from the first 5 DG neurons. The input was introduced to the network up to 5 consecutive times (milliseconds 1-5), where, thanks to STDP, the synaptic weights are modified to learn the spike pattern, also producing an output in the PC neurons corresponding to the stored pattern. The same process is done for patterns 10-11-12-13-14 and 5-6-7-8-9 at milliseconds 15-20 and 29-34, respectively. After learning the patterns during the learning phase, the 1-2-3-4 cue (milliseconds 43-47) is introduced through DG, and the recurrent collateral network of PCs starts to oscillate until the rest of the first pattern (0) is recalled (millisecond 49) reliably (it also appears at milliseconds 45 and 47 while the input sample is being given; therefore, it is not reliable). The oscillatory activity recalling the first pattern is maintained until the arrival of the cue corresponding to the second pattern (10-11-12-13) in milliseconds 57-61, which later allows obtaining the rest of it (14) in millisecond 63. The network continues oscillating after that, but this time recalling the second pattern, and the same process is repeated for the third pattern with cue 5-6-7-8 in milliseconds 71-75, obtaining in millisecond 77 the rest of it (9). As can be seen, once it starts oscillating in a state, it stays in that state until another cue is given. The recalling process has been illustrated with a cue of 4 components of the pattern to recall the fifth component; however, the same behavior would happen when using any combination of any size of the pattern as a cue.

Although the experiments were performed storing up to 4 patterns, the storage capacity of the network is determined by the number of PC neurons and the size of the patterns. Thus, a network with 20 PCs and input patterns of size 4 would be able to store up to 5 orthogonal patterns. For this to work, it is necessary that the learning and recalling phases are separated, since once the first recall cue is used as input, the network starts to oscillate continuously. Therefore, if learning takes place during oscillation, the network will learn the sum of the input patterns together with the current state of the network instead of the original input patterns. Furthermore, for the STDP-based learning process to work correctly, it is necessary to present the pattern several times and achieve constant positive reinforcement on the appropriate connections and negative reinforcement on the others. In this model, it is necessary to only present each of the input patterns 5 times in a row for it to be learned correctly.

Given the architecture of the network, the oscillation between states causes the new state to be reinforced and the previous one to be forgotten when changing from one state to another. Therefore, each pattern can only be remembered once, i.e., the model is a volatile memory. This makes sense from a biological point of view: the hippocampus is a short-term memory, and thus, information is learned and forgotten fairly quickly, being retained longer as long as the network keeps recalling it, or forgotten if the network changes to a different state because the information is no longer useful or simply stored in another memory region after being processed.

Regarding experimentation with non-orthogonal patterns (bottom plot of Fig. 5), the network is able to learn them correctly. However, when it comes to recalling them, due to the oscillatory nature of the network, it will start remembering the pattern correctly, but then the common part with other patterns will be used as a cue of the other patterns that have it. Then, the network will oscillate to a new state of addition of all these patterns, with the consequent learning of this association of patterns and the forgetting of the patterns individually. From a biological point of view, it makes sense in some way, since anterior regions of the hippocampus have the goal of separating patterns through a dispersion of information (increasing the degree of orthogonality), meaning that CA3 also has problems working with non-orthogonal patterns.

B. CA3 memory model with regulated periods of activity

A set of experiments for learning and recalling different combinations between 2 and up to 4 fully orthogonal patterns were carried out on the network. In the upper plot of Fig. 6, the result of the network regarding the recall phase of 3 orthogonal patterns can be observed. For the sake of clarity, the learning process of the patterns in this model has not been included, since it is also similar to the one that can be seen in the experiments that were performed for the oscillatory model. In the recall phase, the weights were fixed to a weight value obtained from the result of the previous training phase
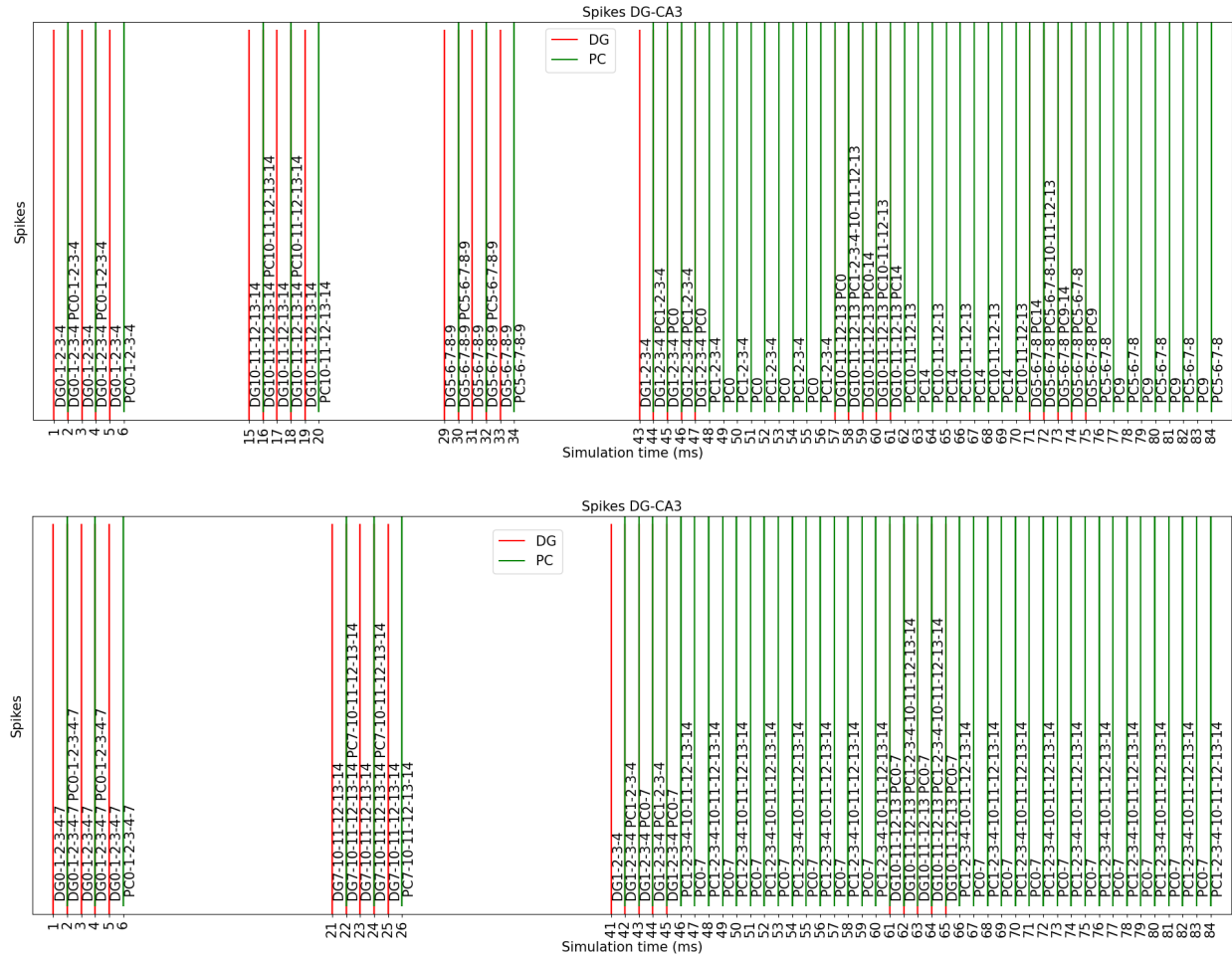
Fig. 5. Results of the experiments performed using the oscillatory network for the learning and recall of 3 orthogonal patterns (top) and 2 non-orthogonal patterns (bottom). During the first half of both simulations, the learning phase of the patterns can be observed, while the second half corresponds to the recall phase of the different patterns.

TABLE I
COMPARISON IN TERMS OF RESOURCES, PERFORMANCE AND FUNCTIONALITY OF BOTH MEMORY MODELS.

Network	Neurons	Static synapses	STDP synapses	Learning (ms)	Recall (ms)	Operations phases	Patterns	Energy consumption	Stability and Reliability
Oscillatory DG-CA3	2*n	n	n*(n-1)	14	14	Both in same simulation	Only orthogonal	High	Low
Regulated DG-CA3	3*n + 1	4*n	n*(n-1)	50	14	One simulation per phase	Both	Low	High

(using STDP). In this case, each pattern only needed to be given to the network a total of 4 times in order to learn it appropriately. The learned patterns were 0-1-2-3, 4-5-6-7 and 8-9-10-11, and the cues provided were 0-1-2, 4-5-6 and 8-9-10. The patterns were recalled correctly (the remaining part of the pattern was obtained for each cue: 3, 7 and 11, respectively) by only giving each cue once to the network. The network did not need any kind of oscillation to be able to return the remaining part of each pattern, but directly recalled the appropriate result, thus, achieving optimum performance in terms of power consumption.

As with the previous model, up to 4 patterns were tested, but the storage capacity of the network depends on the number of PC neurons and the size of the orthogonal patterns.

Regarding the experiments with non-orthogonal patterns,

the network was able to learn and recall them correctly. In the learning phase of the bottom part of Fig. 6, which is not represented in the figure, 2 non-orthogonal patterns were shown 4 times: 1-2-3-4-5-6-7-8-9-10-11-12-14 (all but 0 and 13) and 0-1-2-3-4-5-6-7-8-9-10-11-12-13 (all but 14). After learning, the recall of both patterns was performed using the following cues: 1-2-3-4-5-6-7-8 and 0-1-2-3-4-4-5-6-7-8. After providing the cue containing common parts to both patterns two times, the network was able to correctly recall the remaining part of each of them: 9-10-11-12-14 and 9-10-11-12-13.

Despite working correctly with non-orthogonal patterns, the main problem of this network is that it needs a very precise configuration of the recurrent lateral inhibition weights to perform correctly. This configuration is complex to tune
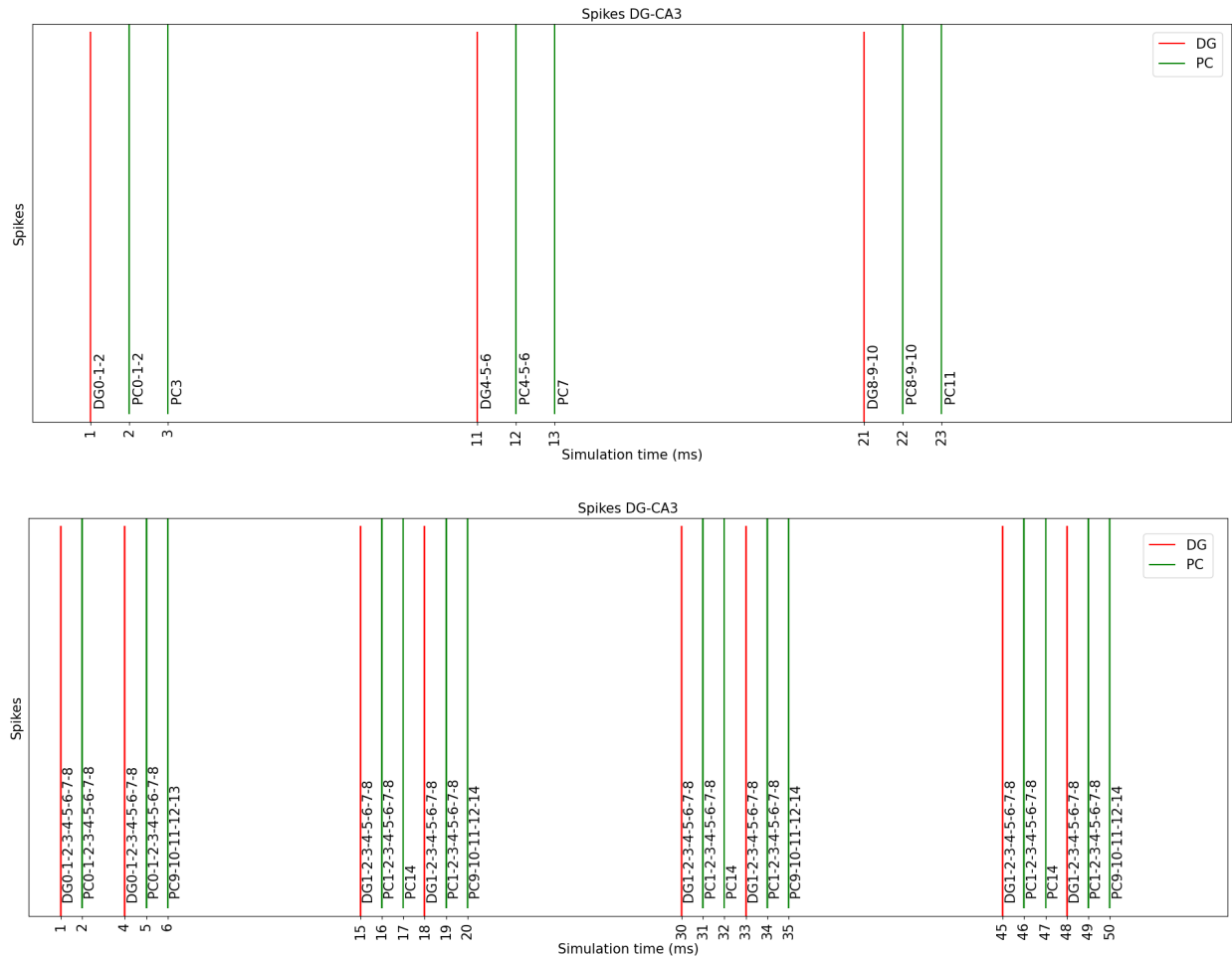
Fig. 6. Results of the experiments performed using the network with regulation of activity periods for the learning and recalling of 3 orthogonal patterns (top) and 2 non-orthogonal patterns (bottom). During the first half of both simulations, the learning phase of the patterns can be observed, while the second half corresponds to the recall phase of the different patterns.

automatically, since it depends not only on the size of the network, the number of patterns to store and their percentage of non-orthogonality, but also on other internal factors of the network such as the resulting weights of the STDP connections, among others, which are difficult to model with an equation. Therefore, the proper use of this model requires a prior configuration step to ensure that it works correctly under the desired conditions.

Thanks to the regulated behavior of the network, which allows having only the necessary activity over time, this model is not only more energy efficient (it only processes the information when there is a new input) but also capable of recalling any internally stored pattern as many times as desired without forgetting it over time or after recalls (which happened in the first model), regardless of whether it is orthogonal or not.

VI. DISCUSSION AND COMPARISON OF THE MODELS

In terms of the architecture and resources, the model with controlled activity is an extension of the first model; therefore, it needs more resources, although both scale linearly as a

function of n (the size of the memory), thus the differences are not substantial between both models, as can be seen in Table I.

In terms of functionality and performance, based on the results of the experiments, the oscillatory model is more bio-inspired due to its constant activity within the attractor network once it starts to recall. However, this also makes it more energetically expensive and unstable as it is constantly generating spikes, even when there is no operation in progress. The second model has additional regulation mechanisms that allow it to maintain minimal activity (only when performing an operation), which makes it more efficient, stable and reliable, since the same patterns can be recalled several times without being erased from memory. On the other hand, these mechanisms make this model have a higher latency in the learning process, as it requires a certain temporal separation between pattern samples. A comprehensive comparison between both models is presented in Table I, where learning and recalling times refer to the time it takes from the start of the operation until the timestep when the next operation can be performed

(not the time that these operations need, which is shorter).

Finally, the first model does not require a prior configuration and it is able to recall any learned pattern immediately after learning it. However, it does not perform well with non-orthogonal patterns, unlike the second model, which is able to work with both kinds of patterns but requires a prior complex parameter setting.

VII. CONCLUSIONS

In this paper, two bio-inspired memory models of the hippocampus with different levels of abstraction have been proposed, implemented and tested. The difference between the models and their biological counterparts allowed exploring alternative ways of storing the information and illustrate the relationship between plausibility and functionality. To the best of the authors' knowledge, the CA3 models presented in this work are the first to be designed and implemented using SNNs on the SpiNNaker hardware platform, not simply simulated in software. Moreover, these models, particularly the second one, are fully functional and completely feasible for their use in real applications based on this spiking paradigm, as they are able to learn and recall patterns of different sizes.

The design of the models and the experiments that were performed illustrate an inverse relationship between plausibility and functionality. Although both are fully functional, the first model is more bio-inspired but is less suitable for a general-purpose domain as it has greater instability, reliability and power consumption, as well as the inability to work with non-orthogonal patterns. Certain changes were needed to make it somewhat less bio-inspired in order to achieve a more functional and useful model for its use in real applications, although this involved having greater learning times.

The strengths and the weaknesses of both models were demonstrated, opening future lines of research in order to study possible improvements. As an example for the first model, the use of collateral recurrent inhibitory interneurons that reduce or even remove the oscillatory activity could be studied, so that it will only remain active for a specific time period when operations are being performed. For the second model, eliminating the need for prior adjustment of the network by modifying or adding connections or populations of neurons could be beneficial, even if the model moves away from plausibility.

The source code regarding the implementation of both hippocampal bio-inspired memories with SNNs on SpiNNaker is available on GitHub[1].

ACKNOWLEDGMENTS

[1]https://github.com/dancasmor/Spike-based-computational-models-of-bio-inspired-memories-in-the-hippocampal-CA3-region-in-SpiNNaker

REFERENCES

[1] C. Mead, "Neuromorphic electronic systems," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1629–1636, 1990.

[2] G. Indiveri, B. Linares-Barranco, T. J. Hamilton, A. Van Schaik, R. Etienne-Cummings, T. Delbruck, S.-C. Liu, P. Dudek, P. Häfliger, S. Renaud *et al.*, "Neuromorphic silicon neuron circuits," *Frontiers in neuroscience*, vol. 5, p. 73, 2011.

[3] K. Roy, A. Jaiswal, and P. Panda, "Towards spike-based machine intelligence with neuromorphic computing," *Nature*, vol. 575, no. 7784, pp. 607–617, 2019.

[4] J. Zhu, T. Zhang, Y. Yang, and R. Huang, "A comprehensive review on emerging artificial neuromorphic devices," *Applied Physics Reviews*, vol. 7, no. 1, p. 011312, 2020.

[5] K. Ahmed, "Brain-inspired spiking neural networks," in *Biomimetics*. IntechOpen, 2020.

[6] J. Wu, Y. Chua, and H. Li, "A biologically plausible speech recognition framework based on spiking neural networks," in *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2018, pp. 1–8.

[7] J. P. Dominguez-Morales, Q. Liu, R. James, D. Gutierrez-Galan, A. Jimenez-Fernandez, S. Davidson, and S. Furber, "Deep spiking neural network model for time-variant signals classification: a real-time speech recognition approach," in *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2018, pp. 1–8.

[8] P. O'Connor, D. Neil, S.-C. Liu, T. Delbruck, and M. Pfeiffer, "Real-time classification and sensor fusion with a spiking deep belief network," *Frontiers in neuroscience*, vol. 7, p. 178, 2013.

[9] B. Rueckauer, I.-A. Lungu, Y. Hu, M. Pfeiffer, and S.-C. Liu, "Conversion of continuous-valued deep networks to efficient event-driven networks for image classification," *Frontiers in neuroscience*, vol. 11, p. 682, 2017.

[10] T. Schoepe, D. Gutierrez-Galan, J. P. Dominguez-Morales, A. Jimenez-Fernandez, A. Linares-Barranco, and E. Chicca, "Live demonstration: Neuromorphic sensory integration for combining sound source localization and collision avoidance," in *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2020, pp. 1–1.

[11] F. Perez-Peña, A. Morgado-Estevez, A. Linares-Barranco, A. Jimenez-Fernandez, F. Gomez-Rodriguez, G. Jimenez-Moreno, and J. Lopez-Coronado, "Neuro-inspired spike-based motion: from dynamic vision sensor to robot motor open-loop control through spike-vite," *Sensors*, vol. 13, no. 11, pp. 15 805–15 832, 2013.

[12] S. Glatz, J. Martel, R. Kreiser, N. Qiao, and Y. Sandamirskaya, "Adaptive motor control and learning in a spiking neural network realised on a mixed-signal neuromorphic processor," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 9631–9637.

[13] D. Gutierrez-Galan, J. P. Dominguez-Morales, F. Perez-Peña, A. Jimenez-Fernandez, and A. Linares-Barranco, "Neuropod: a real-time neuromorphic spiking cpg applied to robotics," *Neurocomputing*, vol. 381, pp. 10–19, 2020.

[14] P. Lopez-Osorio, A. Patino-Saucedo, J. P. Dominguez-Morales, H. Rostro-Gonzalez, and F. Perez-Peña, "Neuromorphic adaptive spiking cpg towards bio-inspired locomotion of legged robots," *arXiv preprint arXiv:2101.09709*, 2021.

[15] E. Donati, G. Indiveri, and C. Stefanini, "A novel spiking cpg-based implementation system to control a lamprey robot," in *2016 6th IEEE International Conference on Biomedical Robotics and Biomechatronics (BioRob)*. IEEE, 2016, pp. 1364–1364.

[16] E. T. Rolls, *Brain computations: what and how*. Oxford University Press, USA, 2021.

[17] C. H. Tan, H. Tang, K. C. Tan, and M. Yuan, "A hippocampus ca3 spiking neural network model for storage and retrieval of sequential memory," in *2013 IEEE Conference on Cybernetics and Intelligent Systems (CIS)*. IEEE, 2013, pp. 134–139.

[18] T. Zhang, Y. Zeng, D. Zhao, L. Wang, Y. Zhao, and B. Xu, "Hmsnn: hippocampus inspired memory spiking neural network," in *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2016, pp. 002 301–002 306.

[19] A. S. Shiva and A. Hussain, "Continuous time recurrent neural network model of recurrent collaterals in the hippocampus ca3 region," in *International Conference on Brain Inspired Cognitive Systems*. Springer, 2016, pp. 343–354.

[20] T. Oess, J. L. Krichmar, and F. Röhrbein, "A computational model for spatial navigation based on reference frames in the hippocampus, retros-

plenial cortex, and posterior parietal cortex," *Frontiers in neurorobotics*, vol. 11, p. 4, 2017.

[21] V. Rajmohan and E. Mohandas, "The limbic system," *Indian journal of psychiatry*, vol. 49, no. 2, p. 132, 2007.

[22] K. S. Anand and V. Dhikav, "Hippocampus in health and disease: An overview," *Annals of Indian Academy of Neurology*, vol. 15, no. 4, p. 239, 2012.

[23] C. G. Wible, "Hippocampal physiology, structure and function and the neuroscience of schizophrenia: a unified account of declarative memory deficits, working memory deficits and schizophrenic symptoms," *Behavioral Sciences*, vol. 3, no. 2, pp. 298–315, 2013.

[24] J. L. Lobo, J. Del Ser, A. Bifet, and N. Kasabov, "Spiking neural networks and online learning: An overview and perspectives," *Neural Networks*, vol. 121, pp. 88–100, 2020.

[25] N. Caporale and Y. Dan, "Spike timing–dependent plasticity: a hebbian learning rule," *Annu. Rev. Neurosci.*, vol. 31, pp. 25–46, 2008.

[26] S. B. Furber, F. Galluppi, S. Temple, and L. A. Plana, "The SpiNNaker project," *Proceedings of the IEEE*, vol. 102, no. 5, pp. 652–665, 2014.

[27] M. Davies, N. Srinivasa, T.-H. Lin, G. Chinya, Y. Cao, S. H. Choday, G. Dimou, P. Joshi, N. Imam, S. Jain *et al.*, "Loihi: A neuromorphic manycore processor with on-chip learning," *Ieee Micro*, vol. 38, no. 1, pp. 82–99, 2018.

[28] P. A. Merolla, J. V. Arthur, R. Alvarez-Icaza, A. S. Cassidy, J. Sawada, F. Akopyan, B. L. Jackson, N. Imam, C. Guo, Y. Nakamura *et al.*, "A million spiking-neuron integrated circuit with a scalable communication network and interface," *Science*, vol. 345, no. 6197, pp. 668–673, 2014.

[29] S. B. Furber, D. R. Lester, L. A. Plana, J. D. Garside, E. Painkras, S. Temple, and A. D. Brown, "Overview of the SpiNNaker system architecture," *IEEE Transactions on Computers*, vol. 62, no. 12, pp. 2454–2467, 2013.

[30] O. Rhodes, P. A. Bogdan, C. Brenninkmeijer, S. Davidson, D. Fellows, A. Gait, D. R. Lester, M. Mikaitis, L. A. Plana, A. G. Rowley *et al.*, "spynnaker: a software package for running pynn simulations on spinnaker," *Frontiers in neuroscience*, vol. 12, p. 816, 2018.

[31] A. P. Davison, D. Brüderle, J. M. Eppler, J. Kremkow, E. Muller, D. Pecevski, L. Perrinet, and P. Yger, "Pynn: a common interface for neuronal network simulators," *Frontiers in neuroinformatics*, vol. 2, p. 11, 2009.

[32] J. Sjöström, W. Gerstner *et al.*, "Spike-timing dependent plasticity," *Spike-timing dependent plasticity*, vol. 35, no. 0, pp. 0–0, 2010.

[33] X. Jin, A. Rast, F. Galluppi, S. Davies, and S. Furber, "Implementing spike-timing-dependent plasticity on spinnaker neuromorphic hardware," in *The 2010 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2010, pp. 1–8.