

Bioinspired Spike-Based Hippocampus and Posterior Parietal Cortex Models for Robot Navigation and Environment Pseudomapping

Daniel Casanueva-Morato,* Alvaro Ayuso-Martinez, Juan P. Dominguez-Morales, Angel Jimenez-Fernandez, Gabriel Jimenez-Moreno, and Fernando Perez-Peña

The brain has great capacity for computation and efficient resolution of complex problems, far surpassing modern computers. Neuromorphic engineering seeks to mimic the basic principles of the brain to develop systems capable of achieving such capabilities. In the neuromorphic field, navigation systems are of great interest due to their potential applicability to robotics, although these systems are still a challenge to be solved. This work proposes a spike-based robotic navigation and environment pseudomapping system formed by a bioinspired hippocampal memory model connected to a posterior parietal cortex (PPC) model. The hippocampus is in charge of maintaining a representation of an environment state map, and the PPC is in charge of local decision-making. This system is implemented on the SpiNNaker hardware platform using spiking neural networks. A set of real-time experiments are applied to demonstrate the correct functioning of the system in virtual and physical environments on a robotic platform. The system is able to navigate through the environment to reach a goal position starting from an initial position, avoiding obstacles and mapping the environment. To the best of the authors' knowledge, this is the first implementation of an environment pseudomapping system with dynamic learning based on a bioinspired hippocampal memory.

1. Introduction

The brain, despite being one of the most complex and less known organs, has demonstrated great computation capabilities, adaptability, and efficient resolution of complex problems, overcoming modern computers in many aspects.^[1] On this basis, neuromorphic engineering focuses on the study, design and implementation of hardware and software systems that mimic the principles of brain structures, and their function to achieve the observed power efficiency and computational capabilities.^[2–4]

In order to develop systems that mimic these biological principles, a bioinspired computational approach is needed. A specific type of biologically plausible neural network, called spiking neural network (SNN), is the most widely used for this purpose. These networks, sets of neurons interconnected by synapses, are responsible


for the generation, processing, and transmission of asynchronous electrical pulses called action potentials or spikes. The transmission of spikes through different layers of neurons allows these systems to achieve high-level functionalities. Within the system, only the set of neurons that receive spikes are active, thus achieving a distributed computing approach that has great advantages in terms of energy consumption and real-time operation compared to traditional systems.^[5,6]

In the neuromorphic engineering field, the navigation of animals and, particularly, mammals, through unknown environments is a topic of great interest due to its applicability to robotics. However, this is still an open challenge to be solved. In biology, the navigation process of mammals consists of different stages: mapping the environment, path planning and decision-making, or motor control.^[7,8] Different brain regions are involved in the navigation process and each of its stages. Among them, the hippocampus and the posterior parietal cortex (PPC) should be highlighted. The hippocampus is the brain region that acts as a short-term memory capable of maintaining an allocentric representation of the environment map thanks to the place cells.^[9,10] On the other hand, the PPC is the region responsible for, among other functions, the planning and modulation of the mammal's movements.^[11–13]

D. Casanueva-Morato, A. Ayuso-Martinez, J. P. Dominguez-Morales, A. Jimenez-Fernandez, G. Jimenez-Moreno
Robotics and Tech. of Computers Lab
Universidad de Sevilla
Sevilla, Spain
E-mail: dcasanueva@us.es

J. P. Dominguez-Morales, A. Jimenez-Fernandez
SCORE Lab
I3US
Universidad de Sevilla
Sevilla 41004, Spain

F. Perez-Peña
School of Engineering
Universidad de Cádiz
Cadiz 11001, Spain

 The ORCID identification number(s) for the author(s) of this article can be found under <https://doi.org/10.1002/aisy.202300132>.

© 2023 The Authors. Advanced Intelligent Systems published by Wiley-VCH GmbH. This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

DOI: 10.1002/aisy.202300132

Biological evidence suggests that the functioning and interconnection of these regions play an important role in the navigation process.^[14] The hippocampus is involved in the exploration of the environment by maintaining an allocentric representation of the state of the environment, while the PPC is responsible for decision-making or translation of this allocentric information from the environment into a sequence of motor actions (egocentric).^[14–16]

Regarding this topic, state-of-the-art works can be found in the literature. Some authors have attempted to develop navigation systems using spiking but not bioinspired networks, refer other studies.^[17] Others propose fully bioinspired navigation systems (including all the stages in the process), such as other studies,^[14] which do not present any learning phase and, therefore, the environment is hard-coded in a static memory,^[8] which is only able to map three different possible states.

Other works focus on a specific stage of the whole process, such as environment mapping,^[15,18–20] path planning,^[7,21–24] and motor control.^[25,26] In general, these works assume that the previous stages are global and return detailed and complete information that is complex to obtain in a spiking paradigm. Moreover, they operate globally rather than locally. Therefore, the mapping is done on the complete environment, the planning generates all the steps to reach the target position from the initial position, and the motor control returns the exact sequence of movements to follow the complete route. These processes require very detailed information about the conditions of the complete environment, and they either are not always available or require information obtained through these same processes but locally.

Given these shortcomings and limitations, this article proposes a fully functional spike-based robotic navigation and environment pseudomapping system implemented with SNN on the SpiNNaker hardware platform. This system is capable of navigating while avoiding obstacles in an initially unknown grid environment to reach a goal position while mapping the environment with several possible states for each position. The operation of this system is achieved thanks to the use of a bioinspired hippocampus memory model capable of generating and maintaining the environment map and a bioinspired PPC model which is responsible for local decision-making or planning.

1.1. Biological Background

1.1.1. Hippocampus

The hippocampus or hippocampal system is that set of brain structures, belonging to the limbic system, which receives information from the different cortical sensory flows from the neocortex. All this information reaches the hippocampus mainly through the EC (Brodmann's area 28) which, at the same time, acts as its main activity output pathway.^[9]

This region is characterized by acting as a short-term memory able to quickly store input information without following any structure. This learning and storage occurs with the association of the different input activity flows by means of an internal self-associative or attractor recurrent collateral network. As the input information is received from different cortical sensory streams, the association of this sensory information forms what

is called a memory or episode. Thus, the hippocampus is involved in episodic memory.^[9,10]

Events stored in the episodic memory lack interpretation or reasoning and are merely associations of information without cognitive processing. It is the semantic memory of the neocortex which gradually builds and adjusts, on the basis of much accumulated information, the semantic or cognitive representation of events.^[9,10]

Furthermore, the hippocampus consists of place cells, which are neurons that are activated when the subject is in a certain spatial region (not a specific point, but a specific area). A network of such neurons in the hippocampus provides a cognitive map of the entire environment in which the individual is located. This map stores an allocentric representation of both the environment and the location of the individual in that environment. It contains updated and coherent information regarding the state of the environment. This means that the hippocampus is involved in the navigation process through the environment.^[9,13,27]

The firing or activation field of the hippocampal place cells or, in other words, the state of the stored map, can be modified by both external signals (new sensory information from the environment) and internal signals (information regarding the individual's own movement).^[13]

1.1.2. Posterior Parietal Cortex

The PPC (Brodmann areas 5, 7, 39 and 40) is an "associative" cortical region, since it is neither strictly sensory nor motor, but combines input information from several brain areas (such as somatosensory, auditory, visual, motor, cingulate, and prefrontal) and integrates proprioceptive and vestibular signals from subcortical areas to achieve higher abstraction functions.^[28]

This anatomical situation within the brain makes it a very active region in several cognitive processes such as: sensorimotor integration, functional memory, imitation of actions observed in other individuals, early motor planning, transformation of sensory information into decisions and actions, and attention and spatial navigation, among others.^[11,28,29] Among all the cognitive processes, its contribution to navigation and the generation of actions based on sensory information should be highlighted.

During the subject's movement in an environment, PPC neurons fire only when a particular movement or action (turning left, turning right, going straight ahead, etc.) is performed at particular (discrete) points along the pathway.^[12,13,28,30] In addition, these firing patterns are modulated from internal information and without the need for external visual stimuli.^[28] These aspects support the fact that the PPC plays a critical and essential role in the transformation of spatial information from landmarks based on allocentric vision (such as that present in the hippocampus) into sequences of movements based on egocentric (first-person) vision, called navigational context, which are necessary to reach the goal position.^[28,30]

Specifically, the activation of PPC neurons encodes the movements to be performed before they take place and is, therefore, responsible for the planning and modulation of the subject's movements.^[11–13] Regarding the internal landmarks, it is worth mentioning the use of the relative position of the subject's head within the allocentric map to determine the movement to be made to reach the target.^[11,28]

The match-selective neurons present in the PPC are responsible for planning the action or intention to be performed. For this purpose, they integrate information from sensory streams together with signals from brain regions involved in more abstract cognitive processes. These signals contain information related to internal brain models that identify the characteristics that are relevant from the point of view of the subject's behavior.^[31] These neurons do not make any decision on the action to be taken until sufficient information is received.^[11]

In short, there are neurons in the PPC that receive the sensory information of the environment and the expected characteristics to carry out a certain behavior; thus, when the activation of both coincide, these neurons are activated, indicating the action to be performed.

These intentions are segregated in the PPC in such a way that each region of the match-selective neurons specializes in planning different actions.^[11] In addition, inhibitory neurons are activated in the PPC between two consecutive decisions^[12] to bring it to a starting state for the following decision.

1.1.3. Hippocampus–Posterior Parietal Cortex Relationship

The availability of a complete map of the environment thanks to the hippocampal system is not enough to perform spatial behaviors. A network capable of converting spatial information into goal-directed motor outputs is needed for this purpose. This is where the PPC, which is in charge of transforming the allocentric representation of the environment stored in the hippocampus and EC into goal-directed movements, comes into play. The connections between the hippocampus and EC with the PPC are responsible for supplying the PPC with the spatial information needed for motor planning.

1.2. Related Work

The navigation process in living beings presents different temporally separated phases of behaviors: environment mapping, path planning and motor control, or translation of the action into motor commands.^[7,8]

Previous works present spiking navigation systems that cover these three phases, such as another study.^[14] In that publication, the authors propose a navigation system based on three spatial frames of reference of the environment (allocentric, egocentric, and route centric) by means of an SNN that consists of models of the hippocampus, the PPC, and the retrosplenial cortex. However, these models are static and do not have any learning capability. Therefore, they only work for completely predefined environments that are previously stored in the models. Furthermore, they are not fully bioinspired models; since, although they mimic the high-level functionality of these brain regions, they do not rely on their neural structure and inner functioning to obtain these higher-abstraction functionalities.

In ref. [8], a complete robotic system based on SNN capable of mapping unknown environments is proposed. Each map position is defined by the activation of a single place cell connected with dynamic synapses to a Border cell in order to detect the presence of obstacles and also to a Goal cell to mark if it is the target. Therefore, the mapping of the environment is limited to three

possible states per position: obstacle, target, and none of the above. It also has other limitations, including the fact that movement planning is limited to avoiding obstacles, and place cells are created on the fly for each new position that is visited. The latter behavior makes the system more optimized for representing the environment, although, at the same time, it moves away from the biological counterpart.

Other works address a particular phase of the navigation using SNN and assume the rest as external inputs/outputs.

For environment mapping, the publication by Nagagawa et al.^[15] proposes a system that stores the set of positions to visit in order to go from the initial position to the target, once it is located. This has also been addressed by other studies,^[18–20] who developed a spiking system capable of, approximately, determining the position of the robot and the presence of obstacles to achieve SLAM.

Regarding the path planning phase, there are numerous works that focus on it. However, most of them assume that the map is static, that it has been completely defined previously and that the map is modeled using a 2D array of neurons, where each neuron represents a position and synapses represent the possibility of going from the source position to the target position.

Other works^[21,22] apply the wavefront technique both in simulation and in real environments to define the path between a source and a target position based on the set of projections that connect both with the smallest delay. A similar implementation can be seen in ref. [23] (goal-driven), in ref. [7] with neurons representing the points of interest instead of specific positions using a winner take all network, or in ref. [24] for dynamic maps with moving obstacles.

Works such as RatSLAM^[32] and NeuroSLAM^[33] are bioinspired in the mammalian spatial cognitive processing and navigational neural systems to develop SLAM systems. Unlike the previous works mentioned, they make use of artificial neural network (ANN) and deep learning techniques for the processing of visual information and odometry data. As a result, these systems are able to map the environment based on landmarks, achieving a more complex, robust, and efficient representation than classical SLAM techniques.

There are variants of these works that make use of the hearing system instead of vision,^[34] applied to 2.5D environments^[35] and even for 3D aquatic environments.^[36]

For the motor control phase, Zahara et al.^[25] proposed a closed-loop control system for a robotic arm with an SNN capable of translating desired actions into motor commands through a prior learning phase with spike-timing-dependent plasticity (STDP), although it is not inspired in any brain region. In ref. [26], the authors design a complete navigation system; however, it focuses on the deployment of self-organizing SNN to generate the appropriate motor command given an input of spatially distributed distance sensors. The system was not able to map the environment in memory and the path planning is based on following an object detected by the distance sensors.

There are also other works^[17] that offer a different approach using SNN, but they are quite distant from their biological counterpart and are rather focused on ANN techniques. It makes use of a convolutional neural network (CNN) network to generate the relevant motor commands via spiking-coded images from datasets.

1.3. Main Contributions

The main contributions of this work include the following: 1) A spike-based bioinspired PPC neural model is proposed to determine and translate the motor action to be performed based on current information regarding the state of the local environment. 2) A robotic system based on SNN capable of navigating a grid environment to reach a target position while avoiding obstacles and mapping the state of the environment in real time is proposed. The system consists of a spike-based bio-inspired hippocampus memory model together with the proposed PPC model. 3) The proposed system is able to store a spiking representation of the map with the travelled path between a source and a target position together with information from the surrounding environment. 4) The system was first simulated in software and then implemented on the SpiNNaker hardware platform and tested in real controlled environments with a mobile robotic platform. 5) The source code is publicly available (see Section 6), together with the documentation including all the necessary details regarding the SNN architectures.

The rest of the article is structured as follows: Section 2 presents the Experimental Section used in this work. In Section 3, the proposed spike-based robotic navigation and environment pseudomapping system is detailed, including its architecture (Section 3.1) and its operating principle (Section 3.2). The experiments performed to evaluate the functionality and performance of the proposed PPC model and the complete system are explained in Section 4, along with the results obtained. Then, in Section 5, the results of the experiments are discussed. Finally, the conclusions of the article are presented in Section 6.

2. Experimental Section

2.1. Spiking Neural Networks

The third generation of neural networks, SNN, is one of the best alternatives to use when working with bioinspired computational models. These allow creating networks of neurons that are inspired in biology and aim to mimic it in order to incorporate the neurocomputational capabilities found in nature.^[37]

The operation and the transmission of information of SNN are based on spikes, which are asynchronous events that incorporate the concept of space and time into neural networks through connectivity and plasticity. These networks are very efficient from a computational point of view, as only the components that are involved in a specific event that occur at a specific moment of time (when input spikes are received) operate, due to their asynchronous behavior.^[38]

At the biological level, there are a wide variety of models for each of the basic components that are present in a neural network depending on the level of abstraction and, therefore, the level of neurocomputational complexity to be achieved. Among them, the most widely used one in the state of the art, the LIF neuron model, was used in this work, which describes the behavior of a neuron by means of an resistor-capacitor (RC) circuit with a spike generator.^[39,40]

On the other hand, STDP learning mechanisms were also used.^[41] The STDP learning mechanism modifies the weight of synapses proportionally to the degree of temporal correlation between the activation of pre- and postsynaptic neurons. When the presynaptic neuron generates a spike before the postsynaptic neuron, the activation of the presynaptic neuron is considered to be related to the activation of the postsynaptic neuron, thus increasing the weight of the synapse that links them. This process is called long-term potentiation (LTP). In the opposite case, that is, the postsynaptic neuron generates a spike before the presynaptic neuron, the activity of the postsynaptic neuron is considered to be unrelated to the activation of the presynaptic neuron. In this case, there is a decrease in the weight of the synapse between them and the process is called long-term depression (LTD). The amount by which the synaptic weight increases or decreases is proportional to the time difference between the two spikes.

From a mathematical point of view, the weight change for an STDP synapse formed by presynaptic neuron j and postsynaptic neuron i , Δw_{ji} , is defined by Equation (1), where t_j^f is the time instant at which presynaptic neuron j produces a spike f , t_i^n is the time instant at which postsynaptic neuron i produces a spike n , and $W(x)$ is the STDP function. The most commonly used STDP function (also used in this work) is defined in Equation (2), where A_+ and A_- identify the maximum amplitude of positive and negative weight change, respectively, and τ_+ and τ_- are the time constants.

$$\Delta w_{ji} = \sum_{f=1}^N \sum_{n=1}^N W(t_i^n - t_j^f) \quad (1)$$

$$W(x) = A_+ \exp\left(-\frac{x}{\tau_+}\right) \text{ for } x > 0$$

$$W(x) = A_- \exp\left(-\frac{x}{\tau_-}\right) \text{ for } x < 0 \quad (2)$$

Different hardware platforms particularly designed for implementing and simulating SNN can be found in the literature. Some of the most well-known hardware platforms are SpiNNaker,^[42] Loihi,^[43] and TrueNorth.^[44]

In this work, we used SpiNNaker as the hardware platform in which the different SNN models presented were implemented and emulated. SpiNNaker is a massively parallel multicore computing system, which was designed to allow modeling very large SNNs in real time. In this work, a SpiNN-5 machine was used, which consisted of 48 SpiNNaker chips in total.^[42,45]

This platform works with a time step of 1 ms time unit and allows the modeling of large SNNs in real time.^[46] In addition, a software package called sPyNNaker^[47] allows running PyNN^[48] simulations directly on the SpiNNaker board, making the platform very straightforward to work with, since all the codes regarding the design and implementation of SNN can be done using high-level functions described in Python programming language.

2.2. Robotic Platform

To test the correct operation of the proposed system in a real, physical environment, a customized robotic platform was used

(Figure 1). This robotic platform is capable of moving through the environment, obtaining information regarding the distance to nearby obstacles in order to detect them and communicating with other systems wirelessly.

To achieve these functionalities, the platform consisted of a circular chassis with two motorized wheels, an idler wheel and an ultrasound sensor mounted on a servomotor to be able to measure the distance to nearby obstacles in different directions around the robot without the need to physically turn. At the electronic level, it contains a Romeo BLE board and an ESP32 board.

The Romeo BLE board is in charge of managing the robot control system. On the one hand, it takes distance measurements on the right, front, and left of the robot through the ultrasound sensor with the help of the servomotor to detect whether or not there are obstacles in the near environment. This information from the local environment is then sent to the spiking system emulated in SpiNNaker for mapping the environment in memory. On the other hand, it has an integrated H-bridge motor controller to perform the motor actions received from the spiking system.

The ESP32 board is connected to the Romeo BLE board and has a wireless communication module. It is responsible for the wireless communication between the robotic platform and the spiking system. It creates an UDP server to which the spiking system must connect in order to transmit information. This allows a fast sending data from the Romeo BLE board to the spiking system and vice versa.

As a wireless robotic platform, it requires a portable power supply. On the one hand, the motors were powered using a set of 4 AAA batteries, while the rest of the electronic systems were powered with an external battery.

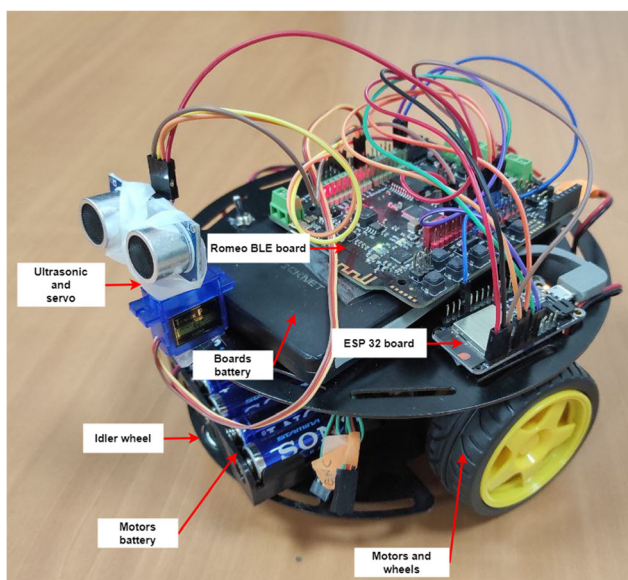


Figure 1. Picture of the robotic platform used for the real-time demonstration. The relevant components are highlighted: the chassis, the power supply system, the robot's control and communication boards, and the sensors and actuators needed to obtain local information from the environment.

3. Environment Pseudomapping and Navigation Model Based on Hippocampus and Posterior Parietal Cortex

This article proposes a spike-based robotic navigation and environment pseudomapping system consisting of a bioinspired hippocampal memory model and a bioinspired PPC decision-making model that are interconnected with each other (see Figure 2). This spiking navigation system follows the biological principles observed in living beings, where the joint functioning of both brain regions plays a critical and essential role in this process, as discussed in Section 1.1.3.

The purpose of the system is to be able to reach a goal position within a grid environment while avoiding obstacles and walls that may be in the trajectory and, at the same time, map the state of the explored fragment of the environment in real time. The hippocampus is in charge of mapping the state of the environment by maintaining a spiking and updated representation of it during exploration, while the PPC is responsible for the decision-making regarding the next action to perform based on the information obtained from the environment that is provided by the hippocampus.

The architecture of the models and, thus, of the navigation system is fully parameterized and depends on the environment to be navigated and the number of possible states that the environment may have. Based on these two variables, the sizes of the different populations as well as the set of internal and external connections and interconnections are automatically calculated.

In the beginning, the environment is completely unknown. The system will map the environment as it is explored and, after reaching the target, maintain a stored representation of the state of the regions it has explored to reach that position. However, it is first necessary to specify the size of the grid in which the environment will be divided. This is necessary to define the possible positions within the environment and, thus, the size of the state map that the hippocampus model will store. The proposed architecture does not limit the maximum grid size.

3.1. Architecture

The complete bioinspired spike-based navigation system is shown in Figure 2. Both the hippocampus model and the PPC model are detailed in Section 3.1.1 and 3.1.2, respectively.

3.1.1. Hippocampus

The bioinspired spike-based hippocampus memory model is based on that reported in our previous work.^[49] This memory model is able to learn, recall, and forget memories following a workflow and structure based on the biological model of the hippocampus. At the input is DG, which is responsible for dispersing input memories and facilitating their storage. Next, DG connects to Cornu Ammonis 3 (CA3) where recurrent collateral synapses with STDP are responsible for learning and storing memories. Finally, CA3 connects to Cornu Ammonis 1 (CA1), which performs the reverse operation of DG to retrieve the memory in its original format before exiting the model.

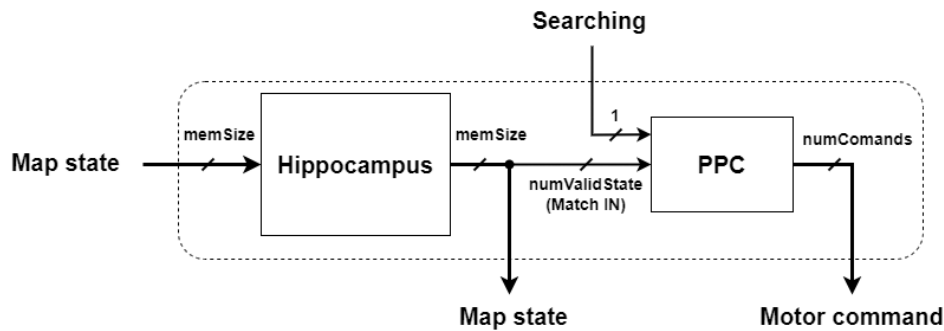


Figure 2. Architecture of the pseudomapping and navigation model proposed. The blocks that correspond to the architecture are framed with a dashed line, which consists of the hippocampus model and the PPC. The architecture is parameterised in size through memSize (size of the memory, equivalent to the number of states plus the number of neurons needed to encode in binary all the positions of the map), numValidState (number of states indicating that the remembered position is traversable), and numCommands (number of motor commands that the PPC can generate). All the interconnections between these blocks are excitatory, static, and parameterized according to the environment in which it will be used.

A memory is considered to be a pattern of spiking activity that is fed to the memory model that consists of concatenation of information from different sources. The information contained in memories lacks prior interpretation, depends on the source it is provided from, and presents a spatiotemporal coding. This means that the activity of different neurons at the same instant belongs to the same memory and that the activity of the same neuron at different instants refers to different memories.

The DG population only disperses a part of the memory, which will be called the cue, while the rest of the memory will be called the content. Because of this division of the memory, CA3 includes CA3 neurons to which the cue arrives (CA3cue) and CA3 neurons to which the content arrives (CA3cont). Recurrent CA3 collateral synapses connecting CA3cue to CA3cont use the STDP learning mechanism. For the implementation of the STDP mechanism, the default values of the hippocampal model have been used. Therefore, A_+ and A_- have a value of 6, and τ_+ and τ_- have a value of 3 time units.

When a memory is to be learnt, it is held at the input to the network for 3 time units, triggering a series of simultaneous activations of CA3cue and CA3cont and, thus, of the STDP mechanism. STDP increases the weight of those synapses that connect the activated neurons or, in other words, learning and storage of the memory occur in the CA3 synapses themselves.

When a memory want to be recalled, a fragment of the memory, the cue, has to be held for one unit of time at the input of the network. The activity of the cue will reach CA3 which, by prior learning of the memory, will activate only those CA3cont neurons belonging to the memory identified by the cue.

In case the memory to be learned has the same cue as an already-stored memory, simultaneous learning and recalling takes place (the learning of the new memory and the recall of the previous one). Due to the temporal sequence of spiking activity that occurs in CA3 as a consequence of these operations, the STDP is activated by increasing the weight of the synapses that represent the new memory and, at the same time, it decreases the weight of the synapses that represent the old one. In other words, at the same time as the new memory is learnt, implicit forgetting of the old one takes place.

Moreover, as in the biological model, since the hippocampus acts as a short-term memory, its content is forgotten over time (memory leak). Using the STDP learning mechanism, for each operation in which a memory is not involved, the weight of the synapses in which it is stored is slightly decreased. Therefore, if after a configurable number of memory operations the memory is not used, the hippocampus model ends up forgetting it, as it is considered not important. Although the forgetting of the memory depends on the number of consecutive operations in which it is not involved, this fact is directly related to the time it has been stored without being used, hence the reason for calling it a temporal forgetting mechanism.

The memories that the hippocampus model learns, recalls, and forgets take a position in the environment as a cue and the state of the environment at that position as the rest of the content. Taking these considerations, the memories would represent maps of the state of the environment in which the subject is located. Following the biological basis, the hippocampus model acts as a short-term memory that, thanks to the use of place cells, is able to maintain an allocentric representation of the map of the environment in which the subject is located.^[14] Place cells would be responsible for storing memory cues.

3.1.2. Posterior Parietal Cortex

The proposed computational model of the PPC is based on the biological basis presented in Section 1.1.2. The characteristics of this region make it an ideal reference point for decision-making,^[50,51] this is, local planning and the generation of the robotic system's movements. By local planning, it is meant that the system decides, based on the information of the local state of the environment, what is the next action to be performed in order to reach the target position.

As shown in **Figure 3**, to perform this task, the PPC takes the time-coded local environmental state information, the combination of information (the state of the local positions) over time during the planning phase, from the hippocampus (Match IN), and a signal that identifies the start of the next decision-making (Searching) operation as input. From this allocentric information of the environment, the model returns the

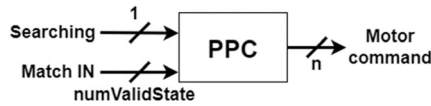


Figure 3. Block diagram of the PPC model including parameterized inputs and outputs. The Searching input signal indicates the start of the motor command planning process whose signal will be propagated internally over time. Match IN are the input signals in which the activation of any of them in conjunction with the Searching signal determines the command to perform, that is, the motor command output.

next motor command (egocentric information) to be performed in order to achieve the goal.

As shown in **Figure 4**, the signal that marks the beginning of the decision-making process (Searching) connects to the population of Delay neurons. This population consists of a set of neurons connected in a chain that is responsible for propagating the input signal with a certain delay. There is a time difference between the start of the decision-making process and the input of the first signal corresponding to the state of the first position of the local environment. As a consequence, the Delay population has an InitDelay input neuron with a different time delay than the rest, which aims to correct this time difference.

Each Delay neuron is connected to a single match-selective neuron. Thus, the population of Delay neurons would act as a selector for a single match-selective neuron for each time window. The time that elapses between the selection of one

match-selective neuron and the next depends on the propagation delay of synapses in the Delay neuron chain.

When the decision-making process starts, the possible states of interest of the different positions of the local environment of the robotic system arrive to the population of match-selective neurons encoded in time. In other words, the network receives a set of inputs that identify the different states that are taken into account in a position to plan the next motor command. With time coding we mean that the activation of these signals at different time instants identifies different positions within the local environment.

The time delay in signal transmission used in the Delay population is the same as the time taken between the temporal state encoding of one position and the next. When the activation of both temporal inputs coincides, a specific match-selective neuron is activated. As each match-selective neuron encodes a single motor command, it is these neurons which, when activated due to the sensory information from the local environment, indicate the action to be performed.

In addition, the model presents a population of interneurons, INH, which inhibit the neurons of the match-selective population and receive an excitatory input from the last neuron of the Delay population. Therefore, the function of these neurons is to regulate the output activity of the model, leaving the match-selective neurons in a resting state between one decision-making and the next, as observed in the biological basis. The rest of the synapses in the model are excitatory.

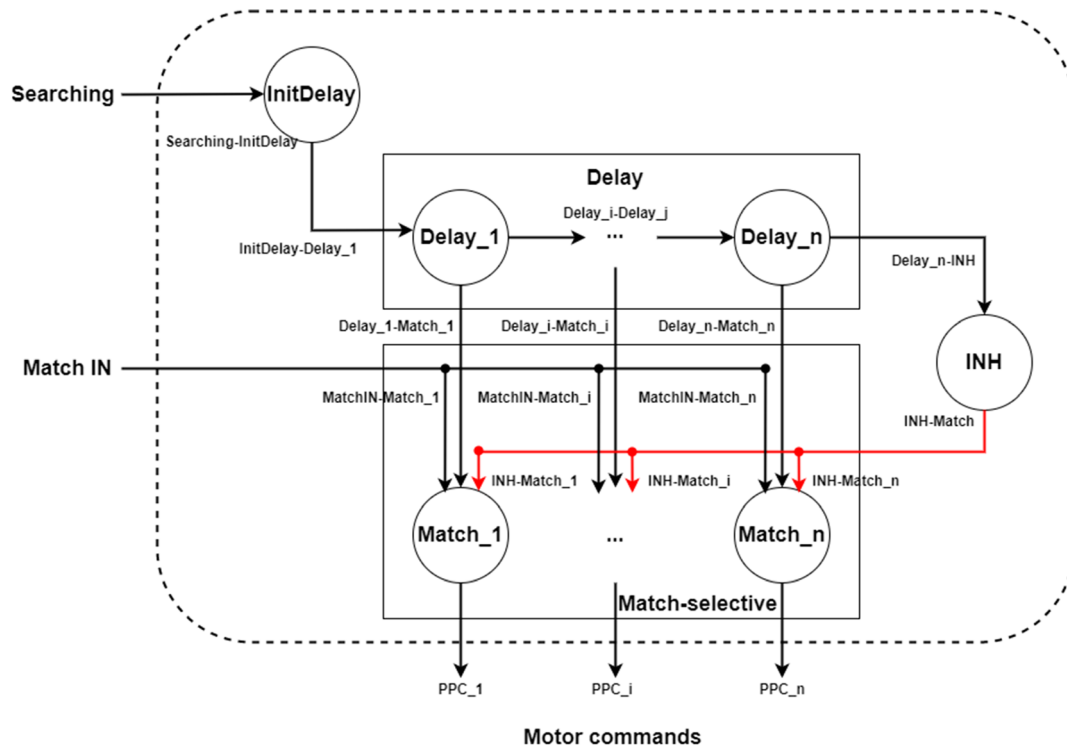


Figure 4. Internal structure of the PPC model. It consists of four populations of neurons: InitDelay, Delay, INH, and match-selective. The InitDelay and Delay populations act as delayed propagators of the input signal, the match population determines whether the Delay activation matches the inputs, indicating the action to be performed, and the INH population is responsible for regulating the output activity of the network. Synapses denoted with black arrows are excitatory and those presented with red arrows are inhibitory.

3.2. Operating Principle

The proposed spike-based robotic system is bioinspired in the navigation system formed by the hippocampus and the PPC. This system enables the navigation of the robot through a grid environment, initially unknown, to reach a goal position avoiding possible obstacles that may appear in the trajectory. At the same time, the system performs a pseudomapping of the state of the environment, allowing to know the current local environment traveled so far in order to determine the actions or motor commands to be performed to reach the goal.

The term pseudomapping is used for two main reasons. The first reason is that the system does not traverse the environment to map it completely: it obtains a map with the state of the fragment of the environment that has been traversed to reach the target position. This map is formed by the combination of the local mapping performed at each position of the trajectory in order to plan the next motor action. The second reason is that the environment is not mapped based solely on the presence of obstacles and targets, as the mapping also considers the input of more complex states obtained by brain systems with higher abstraction functions.

For this purpose, the operation of the system is divided into iterations. In each iteration, the system maps the state of the local environment and, based on this information, determines the next action to be taken. Each of these processes is explained in detail below.

3.2.1. Local Environment Mapping

State Map: The bioinspired hippocampal memory model would be responsible for maintaining the map representation with the state of the environment traversed so far. Each position of the map is represented by the activation of a single PC and, in this model, only one PC is activated simultaneously thanks to the internal mechanism of input information dispersion.^[49] The activation of the PC will be used as a cue to the memory to be learnt and stored, while the remaining content of the memory corresponds to the possible states that these positions may have.

The PC map stores a total of eight possible states per cell. Each PC (i.e., each position on the map) will be associated with the activation of a single state. The main states are: obstacle, goal, free (nonobstacle), start position, and unexplored. The remaining three (step in path, crossroad, and dead end) are derived from these five and are discussed in more detail.

The unexplored status is used to identify those regions that have not been explored yet. The states referring to the presence or absence of obstacles are given by the activity input from boundary cells present in the entorhinal cortex. These neurons are activated when boundaries are detected in the environment either in the form of vertical surfaces or falling edges, that is, in the presence of obstacles.^[52] On the other hand, the goal state is given by the activation of goal cells present in the prefrontal Cortex. These neurons are activated when facing a position that corresponds to the goal.^[8]

The activity of boundary cells and goal cells also reaches other brain regions with higher levels of abstraction functions. In these

regions, information is converted into knowledge, as occurs in the semantic memory.^[9] These areas would be responsible for the activity related to the remaining states that are stored in the positions of the environment map: step in path, crossroad, and dead end.

The step in path state indicates that that specific position is a position that has either been passed through or, after the decision-making phase, established as the next position in the path. The crossroad state identifies a position that has been passed through and where there is at least one additional obstacle-free neighboring position in addition to the one taken in the current path. This state is useful when backtracking during navigation after having encountered a dead end in the path to the goal. Finally, the dead end state indicates which positions that have been traversed lead to dead ends in the trajectory to reach the goal position.

The main states (obstacle, goal, free, start position, and unexplored) are determined internally in the network, while the states derived from them (step in path, crossroad, and dead end) require higher processing. Since the aim of this system is not to compute the most complex states present in biological systems, they are processed externally to the network and fed into the memory afterward.

Implemented Functionality: In the first phase, the system obtains from the real environment the state of the three neighboring positions (right, front and left) with respect to the current position of the robot and stores them in the hippocampal memory. To obtain the neighboring positions, the system combines the orientation and the current position of the robot. The orientation would come from the head direction cells^[52] and the position would come from the hippocampal place cells.^[8]

The possible states of the real environment that reach the hippocampus would be position with obstacle, position without obstacle, and target, as the remaining states would come from further processing. The only exception is the initial position state, which is manually specified in the system in the beginning of the simulation to position the robot within the environment. This information also reaches the PPC where the decision-making phase takes place.

All learning operations applied on the hippocampal memory during the local mapping phase are performed with reinforcement. The hippocampal memory has a forgetting mechanism, which leads to forgetting information that has not been accessed after many consecutive operations. Therefore, as the environment is being navigated, the system will work with nearby locations, while distant ones will be in danger of being forgotten, as they have not been used for several operations. Reinforcement learning consists of a learning operation followed by a recalling operation of the same memory in order to store it for a much longer time.

3.2.2. Decision-Making

In this phase, the PPC performs local decision-making and planning, that is, determining which motor action to perform in order to reach the goal based on the surrounding environment. In this way, the output of the hippocampal memory is connected to the input of the PPC. Specifically, the signals from

hippocampal states of interest that are passed to the match-selective neurons of the PPC are: goal, step in path, and crossroad. This means that the system will determine the motor actions needed in order to reach the positions that present only those states.

By default, the behavior of the system is to search for the next position to reach the target (default behavior). To do so, it obtains the state information of the positions local to the current one from the hippocampus. If any of the neighboring positions present any of the states of interest, the PPC generates the motor action needed to reach it. At the same time, if the current position has more than one neighboring position without obstacle, its state would be updated to crossroad.

If no motor action is generated (no goal, step in path, or crossroad states found in its local environment), a global planning that is external to the network would be called to determine, from all the nonobstacle positions around the current one, the closest to the goal based on the Manhattan distance, and its state would be modified in memory to step in path. In the next iteration of the system, the PPC should find at least one position with a state of interest.

When checking the positions, if it is completely surrounded by obstacles except for the previous position through which it arrived at the current one, it means that it has encountered a dead end. In this case, the system would switch to a backtracking behavior. In this behavior, the system would turn around and start backtracking until it encounters a position with crossroad state. Since it is a crossroad, this position will have at least one other possible neighboring position with no obstacle; thus, the system will again continue to search for a path to the target starting from there. While performing the backtracking, it changes the step in path positions to dead end and the crossroad positions that do not have any unexplored neighboring positions with no obstacles to step in path.

The system will finish its operation when reaching the target position.

4. Experimentation and Results

A set of experiments were performed to demonstrate the correct functioning of the system.

For the case of the memory model, several experiments were performed in a previous work that validated its operation.^[49] In this article, it was implemented with the sPyMem Python library (<https://github.com/dancasmor/sPyMem>).

After that, an incremental experimentation focused on the whole system was performed. These experiments consisted of the navigation and mapping of a set of virtual grids of different sizes and with different distribution of obstacles. In these virtual environments, the robotic system was not used, and, therefore, data inputs from the environment were simulated and given as input to the system.

Initially, the robotic system was evaluated on the navigation and mapping of small environments with few obstacles as a proof of concept. Subsequently, after demonstrating its operation in simple environments, larger environments with more strategically placed obstacles were used in order to test the system in different situations. After deeply evaluating the system in a

virtual environment, a demonstration was performed on a physical environment using the robotic system described in Section 2.

All experimentation on the complete system in both virtual and real environments took place in real time and on a 2D plane. Due to these characteristics of the environment, a total of four motor actions are sufficient to move the robot through it, or, in other words, a total of four match-selective neurons in the PPC model are needed to cover the spectrum of possible motor actions. These actions would be: moving up, down, left, and right.

To determine the motor action to be performed, a recall of the states of the neighboring positions is configured in such a way that the order of the input positions' state to the PPC is: up, left, down, and right. In addition, the PPC model is defined with a delay in the propagation of the onset signal of the decision-making phase equal to the time it takes to perform a recall in the memory model. By taking both considerations, the activation of the four match-selective neurons will indicate the motor actions required for moving up, left, down, and right, respectively.

In addition, for these experiments, a numerical code is used to represent the states of the positions. This number represents the id of the hippocampal memory neuron whose activity encodes that particular state.

4.1. PPC Proof of Concept

This experiment is a proof of concept to demonstrate the correct functioning of the PPC model. For this purpose, a PPC model capable of reacting to two possible input states (INmatchPPCi) and, from them, generating four possible motor actions (Matchi) was developed. This model has a signal propagation delay of 7 ms within the Delay population and 9 ms as the initial delay (InitDelay). In other words, it takes 9 ms from the beginning of the decision-making phase to the arrival of the state of the first position and a time of 7 ms between state arrivals of the remaining consecutive positions.

The test consists of carrying out four decision-makings, in each of which a different action is determined by the arrival of one of the two states of interest. Specifically, the model should select the first (upward), second (left), third (downward), and fourth (right) actions for the first, second, third, and fourth decisions, respectively. Following the analogy discussed at the beginning of the section, the inputs from the states of interest will arrive temporarily in the following order: up, left, down, and right.

Figure 5 shows the input and output spiking activity of the model, as well as the spiking activity of each neuron of each internal population of the network that is activated as a consequence for every time step of the simulation.

The first decision-making (ms 1) starts with the arrival of a spike from the INsearchingPPC0 input. This spike will trigger the activation of the Delay population input neuron, InitDelay, at ms 2. This neuron starts the propagation of the input signal along the Delay neuron chain. The first activation, Delay0, occurs at ms 11 after 9 ms of delay, while the remaining neurons in the chain (Delay1, Delay2, and Delay3) will be activated every 7 ms (ms 18, 25, and 32, respectively). This temporal deviation

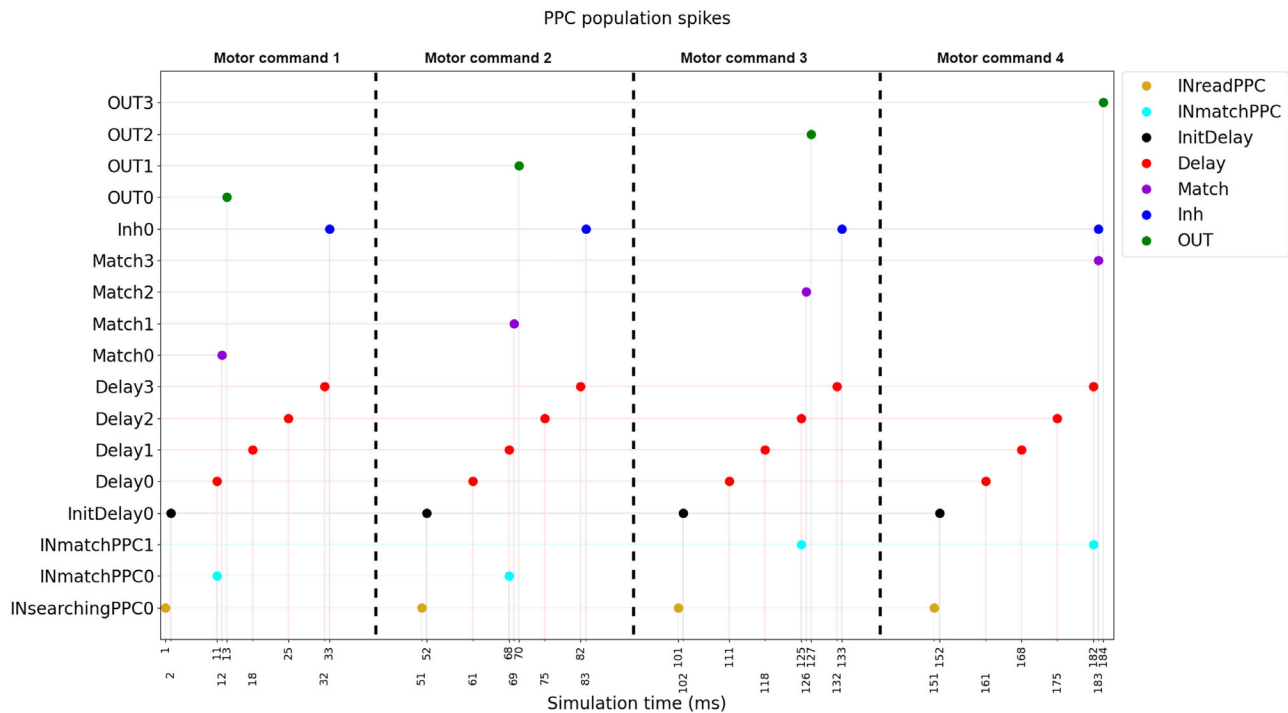


Figure 5. Internal and external spiking activity of the proposed PPC model during a proof of concept test for the four possible motor commands. Each point represents the activation of a neuron (Y axis) at a given time instant (X axis). The plot is divided into four parts, each demonstrating the selection of a different motor command. Each part starts with the activation of the decision-making signal (INsearchingPPC0), which is propagated along each neuron of the Delay population with a certain delay. When the activation of any of the input signals of interest (INmatchPPCi) coincides with the activation of a Delay neuron (Delayj), the corresponding match neuron (Matchj) is activated and, thus, a motor command (OUTj) is selected.

between the activation delay of the first neuron and the remaining neurons is necessary to correct the temporal differences that exist between the beginning of the decision-making process and the arrival of the state of the first position of the local environment.

At the same time, at an instant in the decision-making process, one of the two inputs related to the states of interests for the position being evaluated will be activated. The activation of any of the two is sufficient to select the action to be taken. Specifically, for the first decision-making, the input of the first state of interest, INmatchPPC0, is activated at ms 11. It coincides with the activation of the first neuron of the Delay chain, Delay0. The first match-selective neuron, Match0 is then activated at ms 12. In other words, the action of moving up is selected, whose decision will reach the output population of the model at ms 13.

After the activation of the last neuron of the chain in the Delay population (Delay3) at ms 32, the decision-making phase is finished. Therefore, this activity is then propagated to the inhibitory interneuron Inh0, whose activation at ms 33 will inhibit the match-selective neurons in order to prepare them for the next decision-making.

What was described for the first decision-making occurs for the following ones at ms 51, 101, and 151 respectively. The signal of the first state of interest (INmatchPPC0) is activated for the second decision-making at ms 68 and the signal of the second state of interest (INmatchPPC1) for the third and fourth decision-makings at ms 125 and 182, respectively. This activity

coincides in time with those of the Delay1, Delay2, and Delay3 neurons, respectively, leading to the activation of the Match1 (ms 69), Match2 (ms 126), and Match3 (ms 183) neurons. This activation of match neurons results in the selection of the second action for the second decision-making, the third action for the third decision-making, and the fourth action for the fourth decision-making, as reflected in the population external to the PPC model (OUTi) at ms 70, 127, and 184, respectively.

With this experiment, the correct functioning of the model in the selection of the motor actions for the different possible input conditions in the decision-making process was verified. For each decision-making process, the model was able to generate the expected motor commands after providing the given data inputs.

4.2. 4 × 4 Grid Map Simulation Experiment

This is the first experiment with the complete system on a virtual grid environment of 4 × 4 size, that is, a total of 16 positions distributed in 4 rows and 4 columns. The environment presents a single obstacle strategically placed on the trajectory taken by the robotic system in the ideal case with no obstacles within the same environment (see Figure 6a).

Initially, the system will learn the initial and target positions (Figure 6b). The virtual robot will initially be oriented downwards and, according to this map, start at position 3. After that, the iterative execution of the proposed system will start. In each iteration, the system begins with a local environment mapping phase

in which measurements of the local environment will be taken to determine whether or not there is an obstacle around the current position.

The real-time spiking activity of the system during this phase in the first iteration is shown in Figure 7a. The local environment of the robotic system will be formed by the three surrounding positions, that is, right, front, and left. As the robot starts at position 3 looking down, its immediate right position would be 2, the front would be 7, and the left would be 4. For all of them, the state is an obstacle-free position, as there is only one placed at position 11.

For each position, a state learning operation with reinforcement is performed, that is, a learning operation followed by a recalling operation of the same memory. The memories will consist of the position they encode as a cue and the state of that position as content.

With these considerations in mind, in Figure 7a, the learning and recalling operations of the state of position 2 are observed. Learning is characterized by the activation of neuron 1 of the memory cue population (MemCue1), encoding position 2 in binary, and neuron 5 of the memory content population (MemCont5), encoding the obstacle-free state, at ms 142 and 145. For the recalling operation, the position is passed as a cue (activation of MemCue1) at ms 176 leading to the recall of the state of that position (activation of MemCont5) at ms 177.

The same occurs for the remaining neighboring positions at ms 207–243 for position 7 (MemCue0, 1 and 2) and ms 272–307 for position 4 (MemCue2), with an obstacle-free state or activation of MemCont5.

For the learning operations, the memory must receive the input memory for three consecutive time units. However, only the first and third input stimuli affect the memory; the second input spike is ignored by neurons due to their refractory period. Considering this, for the first operation, for example, if the first spike occurs at ms 142, the next one should be at ms 144. Since the system is working in real time, it may not reach the exact precision of 1 ms between signal transmissions and it may arrive 1 ms later, as in this first learning operation. However, the memory is sufficiently consistent, to ensure that it does not pose a problem and can function correctly even in these cases, as demonstrated by the correct recall of the stored memory during reinforcement learning.

After this, the decision-making phase takes place, in which the next action to be taken to reach the goal position will be determined or, if there is no neighboring position with the step in path state, the most suitable position for being the next in the path will be selected.

In the first iteration, the system is surrounded by free neighboring positions, but none of them has the step in path state (see Figure 6b). By means of Manhattan distance, positions 7 and 2 are the closest to the target. As there are more than one option, by default, the system will choose the position that reduces the distance on the vertical axis, that is, position 7. At the same time, since the current position (position 3) has more than one possible path to the target, its state will change to crossroad. Both learning operations with reinforcement are reflected in Figure 7b.

Both learning operations with reinforcement are shown at ms 474–509 for position 7 with the step in path state (activation of MemCont3) and at ms 539–573 for position 3 with crossroad state (activation of MemCont4). Both state changes lead to forgetting the previous state of those positions in order to learn the new ones, as denoted by their last recall at ms 475 and 540 for position 7 and 3, respectively.

The activation of the PPC outputs at ms 476 and 541 is due to the input activity saturation corresponding to reinforcement learning operations on states of interest, as a constant excitatory input is supplied 2–3 times in a row from which only a single arrival is expected at most.

After that, the first iteration would end and the second one would follow. In its decision-making phase, there is a next position in the trajectory to follow; therefore, the PPC generates the motor action necessary to reach it. As shown in Figure 7c, the phase starts with the recall of the state of the local environment at ms 778. When MemCont3 (step in path state) related to position 7 arrives at the PPC at ms 809, PPC's output 2 is activated at ms 811. This output corresponds to the motor action of moving down (continuing forward) one position with respect to the current one.

In short, in the first iteration, the local environment is mapped and the next position to be visited is decided, while in the second iteration the motor command to reach that position is generated. In other words, with this pair of iterations, a step is taken on the path to the goal.

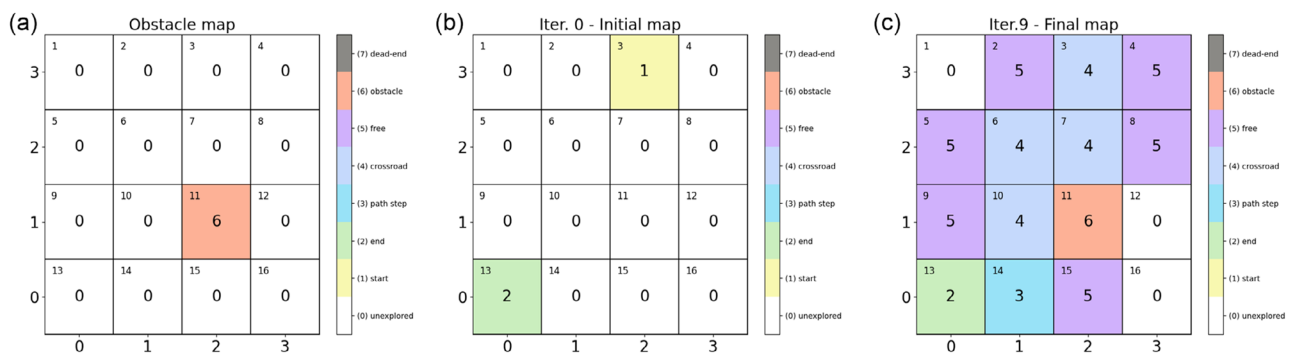


Figure 6. State maps of the environment for the 4×4 grid map experiment: a) showing the location of the obstacle that the system must avoid; b) at the beginning of the simulation with the start and goal positions; and c) after finishing the simulation when reaching the target.

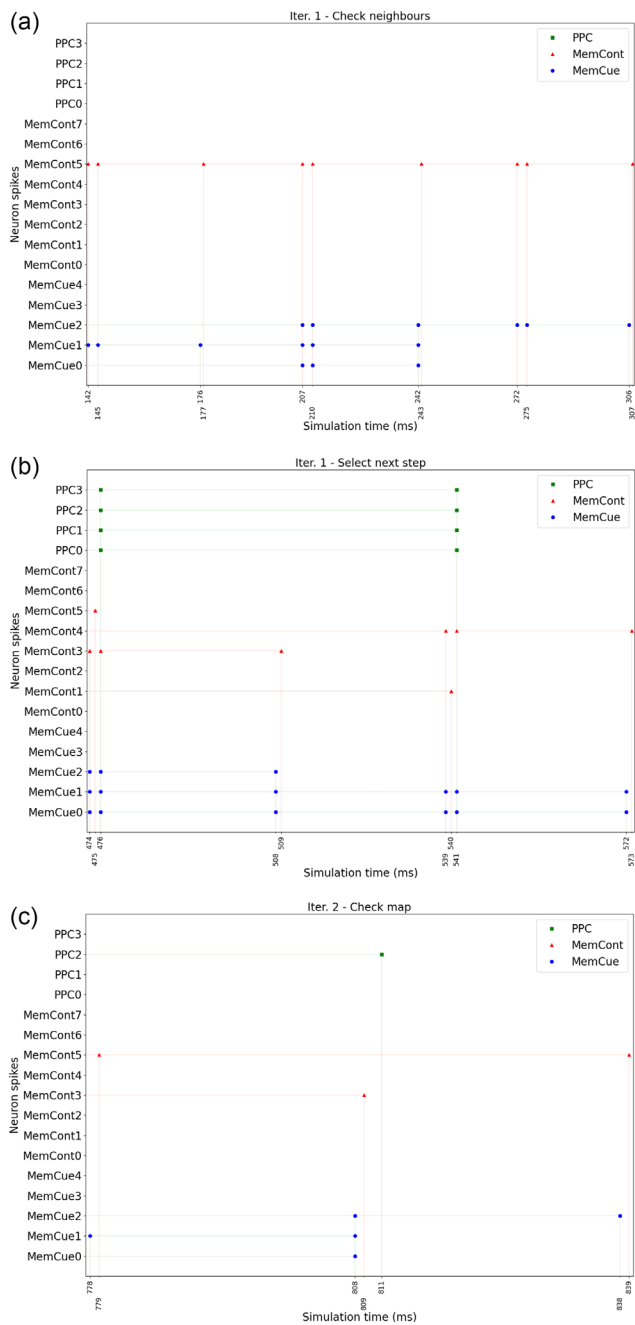


Figure 7. Spike activity of the proposed pseudomapping and navigation model during the 4×4 grid map experiment: a) in a local environment mapping phase during the first iteration; b) when selecting the next position to move to in order to reach the goal during the first iteration; c) in the decision-making phase during the second iteration. Each point represents the activation of a neuron (Y axis) at a given time instant (X axis). The spikes generated by the PPC are shown in green, those generated by the part of the memory in charge of the memory cue are represented in blue, and those corresponding to the part of the memory in charge of the rest of the memory content are displayed in red.

In iterations 3 and 4, the system is at position 7 and detects the obstacle at position 11. This is recorded in the hippocampal memory. The system does not consider that position as the next

step in the trajectory and the PPC ignores that position as it has a state that is not of interest, that is, the position has an obstacle. Therefore, the robotic system successfully avoids it.

From here on, the system will constantly repeat this pair of iterations until the goal is reached at iteration 9. The map of states of the environment stored in the hippocampal memory after reaching the goal can be seen in Figure 6c.

With this experiment, the common operation flow of the complete system is described. At the same time, the correct operation of the system is verified for a 4×4 grid environment with a simple obstacle in the trajectory, being able to travel from the source position to the target position avoiding the obstacle and mapping the environment around the trajectory.

4.3. 6×6 Grid Map Simulation with Short Backtracking Experiment

In this second experiment regarding the complete system, the aim is to test it in a more complex environment than the previous one in terms of size and number and distribution of obstacles. The system will navigate through a virtual grid environment of 6×6 size, that is., 36 positions. In this environment there are a total of six obstacles strategically placed on the trajectory that the robot would take to reach the goal if no obstacles were present. In this case, the obstacles (see Figure 8a) will lead the robot to a dead-end, where a backtracking mode that was not needed in the previous experiment and that allows the robot to go back to the first crossroad and look for another possible path is used.

The system starts the experiment knowing its initial and target position (Figure 8b). Following its default behavior, every two iterations it advances one position in the environment while mapping the state of the neighboring positions. The sequence of positions and actions that the system performs is as follows: position 3 (left), position 4 (right), position 10 (left), position 11 (right), position 17 (left), position 18 (right), and position 24. This way, the robot followed this sort of “ladder movement” to try to reach the goal since, every time it tried to maintain a direction, it encountered an obstacle. During this path, the system mapped positions 5, 12, 23, and 30 with the obstacle state (MemCont6 activation), positions 4, 11, 18, and 24, with the step in path state (MemCont3 activation), positions 3, 10, and 17 with the crossroad state (MemCont4 activation), since they allowed more than one possible path to the goal, and the remaining positions surrounding the followed path that had no obstacles were assigned the obstacle free state (MemCont5 activation). When position 24 is reached, the system does not find any position in the local environment that is free of obstacles, that is, it reached a dead-end. At this point, the system starts its backtracking mode. In this mode, the system turns the orientation of the robot’s head by 180° . In other words, the robot turns around and begins to follow the path it previously took in search for a position with crossroad state. A position with this state means that there is at least one other free position in its local environment that has not been explored yet.

In this way, the system turns around and moves forward through position 18 until it reaches position 17. This backtracking consists of following the positions of the local environment

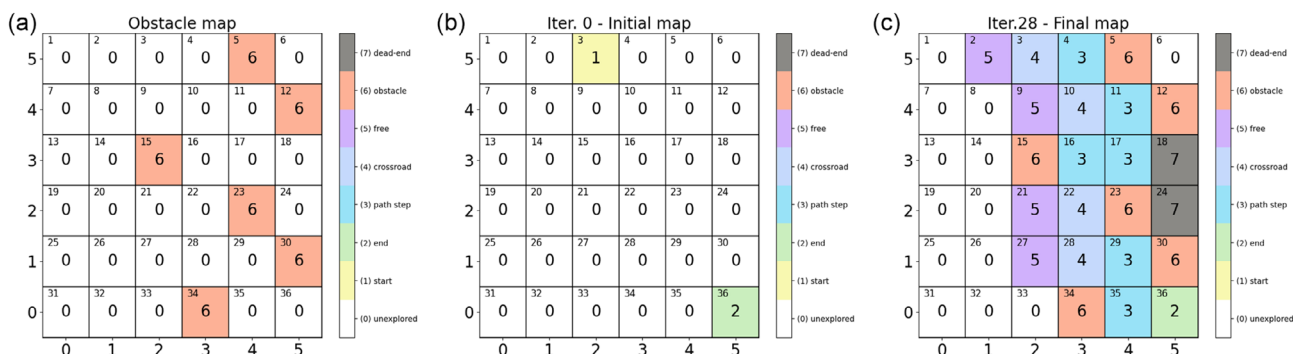


Figure 8. State maps of the environment for the 6×6 grid map with short backtracking experiment: a) showing the location of the obstacle that the system must avoid; b) at the beginning of the simulation with the start and goal positions; c) after finishing the simulation when reaching the target.

with step in path state that are already mapped in the hippocampal map representation. Therefore, to step through each position, only a single iteration is required: in the same iteration the state of the local positions is recalled and the PPC generates the necessary motor action to proceed. At the same time, all the positions traversed until reaching the one with the crossroad state are marked with the dead-end state (activation of MemCont7) to indicate that these positions only lead to dead-end paths.

At position 17, the system returns again to the default behavior mode, continuing its search for the next position toward the target, giving preference to those positions that have not been visited yet. When passing through position 17 again, as it has only one possible position left to continue that was not visited, its state is changed from crossroad to step in path. Following the default mode of operation by pairs of iterations, the system reaches the target position at iteration 28. The state map of the environment stored in the hippocampal memory after the simulation can be seen in Figure 8c. Position 14 is not marked as a crossroad because the system reaches the target position and terminates the execution, and therefore it does not update the position.

This experiment demonstrates the correct behavior of the system in a larger environment and with a more complex obstacle placement, forcing the robot to temporarily switch to backtracking mode, where it stays for a couple of iterations until reaching a crossroad position.

4.4. 6×6 Grid Map Simulation with Long Backtracking Experiment

In this experiment, the system navigates through a virtual environment of 6×6 size with a large number of obstacles than in the previous experiment. These obstacles (see Figure 9a) prevent the system from reaching the goal through the set of ideal shortest paths (right side of the environment). The system goes around these obstacles, traversing almost the entire environment, in order to reach the goal position.

At the beginning, the system follows the same sequence of actions as in the previous experiment. The difference is present when reaching position 17 after backtracking, where it continues its trajectory through position 16 and arrives at position 22, where the system again encounters another dead-end. Unlike in the previous experiment, the backtracking path it has to perform involves more iterations and backtracking movements. Specifically, it completely explores the whole right part of the environment until it returns to the initial position. In this process, the dead-end state is assigned to all the positions in that region.

After that, it takes the path to the left side of the environment despite moving away from the goal, in order to explore other possible paths that will allow the robot to reach it. On this path, it avoids the central obstacle wall until reaching the goal position without further problems. After reaching this position, the

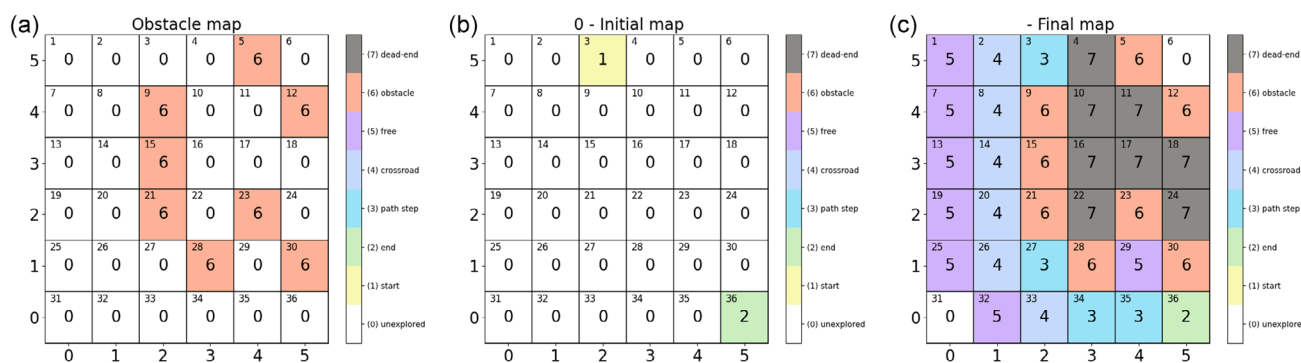


Figure 9. State maps of the environment for the complex 6×6 grid map experiment: a) showing the location of the obstacle that the system must avoid; b) at the beginning of the simulation with the start and goal positions; c) after finishing the simulation when reaching the target.

hippocampal memory presents an almost complete state map of the environment containing the shortest path between the initial position and the goal (see Figure 9c).

This demonstrates the correct operation of the complete system in both default and backtracking behavior, as well as its ability to map the environment, even in complex and relatively large environments.

4.5. Real-Time Demonstration Using a Robotic Platform

The previous experiments demonstrated the operation of the system in virtual environments, simulating the inputs and outputs. This experiment was performed in order to prove the operation of the whole system in a real scenario. A real-time demonstration of the system operation in a real physical environment was developed using the robotic platform described in Section 2.

For this experiment, the same grid as the one used in the “4 × 4 grid map simulation experiment” was used. Therefore, the same initial and goal positions in a 4 × 4 grid environment and a single obstacle placed in the path of the robot toward the goal were used (see Figure 10). In order to prepare this environment, the physical space was divided into cells of 30 × 30 centimeters, resulting in an environment of 120 × 120 cm.

The SpiNNaker hardware platform is used to run the SNN formed by the hippocampal memory and the PPC. The Next Unit of Computing (NUC), an Intel’s mini-computer with large computing capabilities and small size, acts as the core system to launch the network simulation on SpiNNaker and manage the iterative execution of the system and the communication

between the network and the robotic platform. This communication consists of passing the information related to the presence or absence of an obstacle in the local environment obtained from the robot to the SNN implemented in SpiNNaker and the response from SpiNNaker with the motor action generated by the PPC to the robot in order to execute it.

After the experiment was finished, the robotic platform was able to reach the target position while avoiding the obstacle. The spiking system generated exactly the same spiking activity and map as in the 4 × 4 grid virtual environment that used the same configuration. In short, the operation of the system was demonstrated not only in ideal virtual environments, but also in physical and real case environments.

A recording of this experiment is available, with annotations of what happens at each moment. This video can be found in the repository indicated at the end of Section 6.

5. Discussion

The set of experiments performed demonstrated the decision-making capacity of the PPC model to generate the motor actions from the state inputs of the different positions in the local environment. On the other hand, they proved the operation of the complete system in different environments, both virtual with simulated inputs and physical with a robotic platform in a real case scenario. In addition, these experiments were used to test the system progressively in environments whose complexity ranged from lower to higher, varying the size of the environment and the number and disposition of the obstacles in it.

Each experiment on the complete system was used to explore different situations where its ability to adapt to the environment was demonstrated, both in navigation to reach the goal by detecting and avoiding obstacles and in the mapping of the explored environment. In the first experiments, since the environments used were simple, the system barely explored and mapped half of it. On the other hand, in the last experiment consisting of a virtual environment, the system mapped almost the entire environment due to its level of complexity and the need to search for alternative paths to reach the goal position.

The hippocampal memory model shows a certain degree of forgetting for each operation performed (memory leak).^[49] This forgetting factor could be adjusted, by modifying the STDP parameters, to add a temporality factor to the validity of the information. In this way, local and more recent information would be retained for longer, while information from more distant positions in both time and space would be forgotten, that is, they would lose validity. This mechanism would allow the system to reinforce its performance in dynamic environments. When navigating through already visited positions that were forgotten, the system will map them again.

From a bioinspired point of view, the biological basis of the PPC (Section 1.1) describes in broad outline the functions in which it participates, as well as some of the internal structures that form it. When designing and developing the proposed PPC model, we tried to make it as close to these bases as possible.

In the biological model, the PPC actively participates in the navigation process by transforming a sequence of allocentric information from the hippocampus into a sequence of egocentric

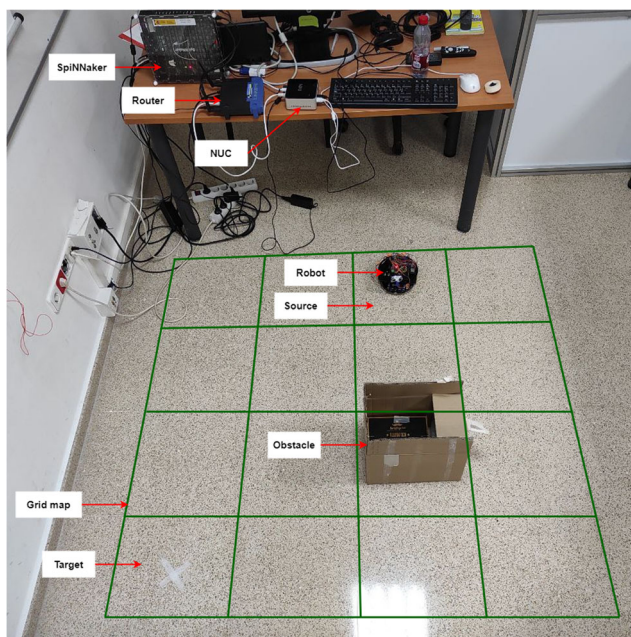


Figure 10. Environment used for the real-time demonstration with the robotic platform. It contains the grid map with the source and target positions of the simulation, the obstacle, and the necessary hardware systems. These hardware systems are the robotic platform, the SpiNNaker board, the computer (NUC) that keeps the whole system running, and the router needed to interconnect the computer with SpiNNaker.

information, such as action sequences. On the other hand, the proposed model takes the information of the state of the local environment (i.e., the surrounding positions) as input to determine the egocentric motor action (turn right, turn left, up and down) to be performed in each iteration. In addition, the local environment state information comes from an allocentric representation of the environment via a map stored and managed by the hippocampal memory model.

The PPC model generates motor actions at each iteration (or pair of iterations), where each iteration represents the location of the system at a different position within a discrete environment. In other words, the PPC generates motor actions at discrete positions as in the biological model.

At the structural level, in the biological model of the PPC, match-selective neurons can be found. Each set of these neurons is responsible for generating a specific motor action based on the information coming from sensory flows and the flow of brain regions with more abstract cognitive processes. Similarly, the proposed model presents a population of match-selective neurons, each of which is responsible for generating a specific motor action. This population determines the action to be performed based on the information from the states of interest of the local positions where the system is located. States derive directly from external sensory information or from the processing of these.

Although the state of the positions directly come from the map constructed in the hippocampal memory, the information with which these states are assigned comes both from the sensory flow (absence or presence of obstacles, target, etc.) and from regions with more complex cognitive processes (step-in-path, dead-end, etc.). In addition, the proposed model presents a population of inhibitory interneurons that are responsible for resetting the state of the match-selective neurons (which is also present in the biological model) after each iteration.

The biological model of the hippocampus indicates that it acts as a short-term memory and is also capable of maintaining an allocentric representation of the environment. This representation is achieved by place cells, which are neurons that are activated when the mammal is in a certain position in the discretized environment. In the proposed system, the hippocampus acts as a memory capable of storing memories that form an allocentric representation of the environment. This is achieved by considering that memory cues encode the activity of the place cells (the positions in the environment) and that the content represents the possible states of those positions.

As discussed in Section 3.2.1, the states of positions come from neurons and brain regions whose activity is received at the hippocampus. The states related to the presence of obstacles, absence of obstacles, initial position, and goal come from the border and goal cells, while the states of step in path, crossroad, and dead-end would come from those brain regions involved in more abstract processes, such as semantic memory.

The complete system presents a default behavior when searching for new positions that bring it closer to the goal and a backtracking behavior in which it searches for new possible paths for reaching the goal position. This has also been observed in biology. Authors in ref. [13,14] indicate that the functioning of the PPC and the hippocampus is sensitive to the general behavior of the individual. It is therefore plausible that, during navigation,

depending on the individual's situation, one behavior or another is adopted in these brain regions.

However, there are some aspects of the model that are not similar to its biological counterpart. There is a set of features yet to be implemented by SNN that are beyond the scope and purpose of this work. The mapping of the local environment and the local decision-making or path planning are performed by the proposed spiking network. The management signals of the system iterations, the global planning (deciding which is the closest position to the goal by Manhattan distance of those found with the local environment mapping), and the communication with the robot are external to the spiking network.

The bioinspired nature of the proposed system comes from its inspiration in the brain regions of the hippocampus and the PPC, not in the application of ANN-to-SNN conversion methods, as in ref. [17] The proposed system has not only been simulated in software and on ideal environments as in ref. [8,15], but also on more complex environments and, in addition, on a robotic platform in physical environments.

Regarding the environment, the proposed system initially knows nothing about the environment except the initial and goal position. Throughout the simulation, the system maps the environment through a continuous learning process. Therefore, it does not work with completely known and previously mapped environments where no learning is involved, as in ref. [14,24], or those that focus only on path-planning and, therefore, assume that they can access the status of the entire map, such as in other studies.^[7,21–23]

The hippocampus model is responsible for cumulatively storing the result of the local mapping, resulting in a pseudomapping of the environment, while in refs. [26] the mapping result is not stored, but immediately used for local planning. This map presents different states beyond the presence of obstacles or not and the goal position, as in refs. [8,15,18]. In addition, the system reacts to the environment by avoiding obstacles and reaching the goal, not merely circling around the environment avoiding obstacles or even ignoring them as in refs. [8,15].

The proposed system does not present a global planning algorithm achieved with SNNs like those proposed in ref. [7,21–24]. However, these techniques necessarily require knowledge of the entire environment and additional information that is sometimes difficult to obtain. In ref. [18–20] the authors propose SNN models to approximately determine the position of the robot in each iteration of the system to achieve a SLAM, while the proposed system requires external signals to control the position in each iteration. Moreover,^[8,15,18] a more bioinspired position state input system than that of the proposed system is used.

Finally, the system presents two time-dependent tasks. On the one hand, when the state information of the robot's position reaches the hippocampus, the input data to the hippocampus must remain in the input for three units of time. However, in case the data cannot be kept three time units in a row at the hippocampal input, the hippocampus model allows up to five time units difference between two consecutive inputs. In case of larger time differences, the behavior of the system may differ from the expected, as the learning of the memory (state of the environmental locations) will result in an activation of the STDP with a lower synaptic weight variation. This implies the possibility that the memory may not be learnt well or may only be partially learnt.

On the other hand, when the state information of the local positions reaches the PPC, given a specific time configuration, it is expected that the time constraints mentioned before will be respected. That is, given an initial time of x time units until the arrival of the first state and a time between subsequent consecutive state inputs of γ time units, the PPC will function correctly when the inputs reach it at these times. However, based on the configuration of the neural parameters of the PPC model, it allows this input to be delayed or advanced by up to three time units from the expected time.

Despite all of the above, these temporal error margins can be extended, making the robot more fault tolerant. To do so, it is sufficient to modify the configuration parameters of the STDP mechanism in the hippocampus model, for the first case, and of the neuron models in the PPC, for the second case.

6. Conclusion

In this work, a fully functional spike-based robotic navigation and environment pseudomapping system implemented with SNNs on the SpiNNaker hardware platform was proposed. The proposed system is able to navigate through a discrete and initially unknown grid environment in order to reach a goal position from an initial position while avoiding possible obstacles that may be present in the trajectory. Furthermore, the system is able to map the state of the explored environment during its trajectory to the goal.

The architecture of the proposed system is fully parameterized, thus allowing its use in environments of different sizes or even in environments of equal size but with higher or lower spatial resolution in the discretization of the environment. Specifically, it demonstrated good performance for environments of up to 36 positions distributed in a 6×6 grid with many obstacles, but it would work just as well for other maps of different sizes, as well as different number and distribution of obstacles.

In this work, a fully functional bioinspired PPC neural network model implemented with SNNs was also proposed. This model is able to perform local decision-making or planning based on the state of the positions in the local environment. This information regarding the state of the environment is provided by a hippocampal memory model proposed in a previous work.

This memory model is able to maintain a representation of the state of the different positions that make up the grid-like environment through which the system navigates. To do this, it learns and recalls memories that consist of a cue (fragment from which the rest of the content of the memory is recalled) and the content of the memory itself. Each memory encodes the activation of a place cell (i.e., a position on the map of the environment) through the cue, while the rest of the content of the memory indicates the state of that position. The information with which the possible states are established comes from the border cells, goal cells, and other regions with higher abstraction functionalities.

The hippocampal memory system is able to obtain a new form of representation of the environment. It is a spiking and thus dynamic representation of the environment, a spiking map. Each position and the state of each position in the environment is represented by a specific spike pattern. The information is encoded both temporally and spatially, thus allowing new ways of processing information from the environment that take

advantage of the intrinsic characteristics of these types of applications.^[7,53]

Furthermore, the proposed complete system was not only compared with its biological counterpart, but also with other systems that can be found in the literature. A bioinspired spiking system capable of mapping the environment that the robot navigates with a wide range of possible states for each position was proposed. Thanks to this range of states, the system was able to navigate complex environments by detecting and avoiding obstacles until reaching the goal position. In addition, a PPC model capable of performing local decision-making or planning based on information about the state of the local environment was presented.

The proposed functionalities of both the complete system and the bioinspired PPC model were demonstrated through a set of experiments. These experiments were successfully completed by demonstrating, on the one hand, the decision-making capability of the PPC model and, on the other hand, the ability of the complete system to navigate and map a set of progressively larger and more complex environments. Furthermore, its operation was not only simulated, but also tested in a physical environment with a robotic platform in a real-time demonstration.

As a future work, some spiking features of the system could be implemented. The global planning part of the system was not within the scope of this work. Therefore, one of such features would be the adaptation of the wavefront technique for the hippocampal memory architecture as a spiking global planning algorithm. This allows the system to react to dynamic environments by having a global planning that determines the trajectory to the goal and a local planning that rectifies depending on the obstacles it detects. Another interesting feature would be the incorporation of a population of head cells with which to determine in a spiking manner the position in which the system is at any given moment and, therefore, which are the neighboring positions.

In addition, the spiking map generated in hippocampal memory after reaching the goal has many applications in the field of neuromorphic robotics. It could be shared with other spiking robotic systems that, without the need for learning and mapping the environment, could navigate from the source position to the goal or calculate the most optimal route to the goal. Several robots in parallel could explore different regions of an environment and, after reaching their respective goals, assemble a joint spiking representation of their explored regions. Ultimately, the generation of the spiking state map of the environment and even the map itself serve as the basis for spiking robotic swarms.

The source code regarding the implementation of the complete spike-based robotic navigation and environment pseudomapping system with SNN on SpiNNaker is available on GitHub (<https://github.com/dancasmor/bio-inspired-spike-based-hippocampus-and-posterior-parietal-cortex-robotic-system-for-pseudo-mapping>).

Acknowledgements

This research was partially supported by the CHIST-ERA H2020 grant SMALL (CHIST-ERA-18-ACAI-004, PCI2019-111841-2/AEI/10.13039/501100011033) and by project PID2019-105556GB-C33 funded by MCIN/AEI/10.13039/501100011033. D. C.-M. was supported by a "Formación de Profesorado Universitario" Scholarship from the Spanish Ministry of Education, Culture and Sport.

Conflict of Interest

The authors declare no conflict of interest.

Data Availability Statement

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

Keywords

environment state maps, hippocampus, neuromorphic engineering, posterior parietal cortex, spatial navigation, spiking neural networks, SpiNNaker

Received: March 15, 2023

Revised: July 20, 2023

Published online:

- [1] G. E. Budd, *Philos. Trans. R. Soc., B* **2015**, 370, 20150037.
- [2] C. Mead, *Proc. IEEE* **1990**, 78, 1629.
- [3] 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, J. Schemmel, G. Cauwenberghs, J. Arthur, K. Hynna, F. Folowosele, S. Saighi, T. Serrano-Gotarredona, J. Wijekoon, Y. Wang, K. Boahen, *Front. Neurosci.* **2011**, 5, 73.
- [4] W. Gerstner, R. Ritz, J. L. Van Hemmen, *Biol. Cybern.* **1993**, 69, 503.
- [5] K. Roy, A. Jaiswal, P. Panda, *Nature* **2019**, 575, 607.
- [6] J. Zhu, T. Zhang, Y. Yang, R. Huang, *Appl. Phys. Rev.* **2020**, 7, 011312.
- [7] S. Koul, T. K. Horiuchi, *IEEE Trans. Circuits Syst. I, Regul. Pap.* **2019**, 66, 1544.
- [8] G. Tang, K. P. Michmizos, *Gridbot: An Autonomous Robot Controlled by a Spiking Neural Network Mimicking the Brain's Navigational System*, Association for Computing Machinery, New York, NY **2018**, pp. 1–8.
- [9] E. T. Rolls, *Brain Computations: What and How*, Oxford University Press, Oxford, UK **2021**.
- [10] K. S. Anand, V. Dhikav, *Ann. Indian Acad. Neurol.* **2012**, 15, 239.
- [11] D. Lyamzin, A. Benucci, *Neurosci. Res.* **2019**, 140, 14.
- [12] R. A. Andersen, C. A. Buneo, *Annu. Rev. Neurosci.* **2002**, 25, 189.
- [13] J. R. Whitlock, R. J. Sutherland, M. P. Witter, M.-B. Moser, E. I. Moser, *Proc. Natl. Acad. Sci. USA* **2008**, 105, 14755.
- [14] T. Oess, J. L. Krichmar, F. Röhrbein, *Front. Neurobot.* **2017**, 11, 4.
- [15] H. Nakagawa, K. Tateno, K. Takada, T. Morie, *IEEE Access* **2022**, 10, 43003.
- [16] A. Tosoni, M. Corbetta, C. Calluso, G. Committeri, G. Pezzulo, G. Romani, G. Galati, *Eur. J. Neurosci.* **2014**, 39 1370.
- [17] K. D. Fischl, G. Tognetti, D. R. Mendat, G. Orchard, J. Rattray, C. Sapsanis, L. F. Campbell, L. Elphage, T. E. Niebur, A. Pasciaroni, V. E. Rennoll, H. Romney, S. Walker, P. O. Pouliquen, A. G. Andreou, in *2017 51st Annual Conf. on Information Sciences and Systems (CISS)*, IEEE, Piscataway, NJ **2017**, <https://doi.org/10.1109/CISS.2017.7926179>.
- [18] R. Kreiser, A. Renner, Y. Sandamirskaya, P. Pienroj, in *2018 IEEE/RSJ Inter. Conf. on Intelligent Robots and Systems (IROS)*, IEEE, Piscataway, NJ **2018**, pp. 2159–2166.
- [19] R. Kreiser, G. Waibel, N. Armengol, A. Renner, Y. Sandamirskaya, in *2020 IEEE Inter. Conf. on Robotics and Automation (ICRA)*, IEEE, Piscataway, NJ **2020**, pp. 6134–6140.
- [20] G. Tang, A. Shah, K. P. Michmizos, in *2019 IEEE/RSJ Inter. Conf. on Intelligent Robots and Systems (IROS)*, IEEE, Piscataway, NJ **2019**, pp. 4176–4181.
- [21] S. Koziol, S. Brink, J. Hasler, *IEEE Trans. Very Large Scale Integr. VLSI Syst.* **2014**, 22, 2724.
- [22] T. Hwu, A. Y. Wang, N. Oros, J. L. Krichmar, *IEEE Trans. Cognit. Dev. Syst.* **2017**, 10, 126.
- [23] J. Friedrich, M. Lengyel, *J. Neurosci.* **2016**, 36, 1529.
- [24] M. Sakurai, Y. Ueno, M. Kondo, in *2021 IEEE Inter. Conf. on Intelligence and Safety for Robotics (ISR)*, IEEE, Piscataway, NJ **2021**, pp. 209–215.
- [25] O. Zahra, S. Tolu, D. Navarro-Alarcon, *Bioinspiration Biomimetics* **2021**, 16, 036008.
- [26] E. Nichols, L. McDaid, N. Siddique, *Int. J. Neural Syst.* **2010**, 20, 501.
- [27] K. Anand, V. Dhikav, *Ann. Indian Acad. Neurol.* **2012**, 15, 239.
- [28] J. R. Whitlock, *Curr. Biol.* **2017**, 27 R691.
- [29] Y. Zhou, D. J. Freedman, *Science* **2019**, 365, 180.
- [30] D. A. Nitz, *Neuron* **2006**, 49, 747.
- [31] G. Ibos, D. J. Freedman, *Elife* **2017**, 6, e23743.
- [32] M. J. Milford, G. F. Wyeth, D. Prasser, in *Proc. 2004 IEEE Inter. Conf. on Robotics and Automation (ICRA'04)*, IEEE, Piscataway, NJ **2004**, Vol. 1, pp. 403–408.
- [33] F. Yu, J. Shang, Y. Hu, M. Milford, *Biol. Cybern.* **2019**, 113, 515.
- [34] J. Steckel, H. Peremans, *PLoS One* **2013**, 8, e54076.
- [35] M. Milford, D. McKinnon, M. Warren, G. Wyeth, B. Upcroft, *Feature-based Visual Odometry and Featureless Place Recognition for SLAM in 2.5D Environments*, Australian Robotics and Automation Association (ARAA), Sydney, NSW **2011**, pp. 1–8.
- [36] L. Silveira, F. Guth, P. Drews Jr, P. Ballester, M. Machado, F. Codevilla, N. Duarte-Filho, S. Botelho, *IFAC-PapersOnLine* **2015**, 48, 212.
- [37] K. Ahmed, in *Biomimetics*, IntechOpen, London, UK **2020**, chap. 5.
- [38] J. L. Lobo, J. Del Ser, A. Bifet, N. Kasabov, *Neural Networks* **2020**, 121, 88.
- [39] A. Tavanaei, M. Ghodrati, S. R. Kheradpisheh, T. Masquelier, A. Maida, *Neural Networks* **2019**, 111, 47.
- [40] R. B. Stein, *Biophys. J.* **1965**, 5, 173.
- [41] J. Sjöström, W. Gerstner, *Scholarpedia* **2010**, 5, 1362.
- [42] S. B. Furber, F. Galluppi, S. Temple, L. A. Plana, *Proc. IEEE* **2014**, 102, 652.
- [43] M. Davies, N. Srinivasa, T.-H. Lin, G. Chinya, Y. Cao, S. H. Choday, G. Dimou, P. Joshi, N. Imam, S. Jain, Y. Liao, C.-K. Lin, A. Lines, R. Liu, D. Mathaikutty, S. McCoy, A. Paul, J. Tse, G. Venkataramanan, Y.-H. Weng, A. Wild, Y. Yang, H. Wang, *IEEE Micro* **2018**, 38, 82.
- [44] 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, B. Brezzo, I. Vo, S. K. Esser, R. Appuswamy, B. Taba, A. Amir, M. D. Flickner, W. P. Risk, R. Manohar, D. S. Modha, *Science* **2014**, 345, 668.
- [45] S. Furber, P. Bogdan, in *Spinnaker-A Spiking Neural Network Architecture*, Now Publishers, Delft, Zuid-Holland Netherlands **2020**.
- [46] A. Date, E. Bienenstock, S. Geman, *On the Temporal Resolution of Neural Activity, Technical Report*, Brown University, Division of Applied Mathematics, Providence, RI **1998**.
- [47] O. Rhodes, P. A. Bogdan, C. Brennkmeijer, S. Davidson, D. Fellows, A. Gait, D. R. Lester, M. Mikaitis, L. A. Plana, A. G. D. Rowley, A. B. Stokes, S. B. Furber, *Front. Neurosci.* **2018**, 12, 816.
- [48] A. P. Davison, D. Brüderle, J. Eppler, J. Kremkow, E. Müller, D. Pecevski, L. Perrinet, P. Yger, *Front. Neuroinf.* **2009**, 2, 11.
- [49] D. Casanueva-Morato, A. Ayuso-Martinez, J. P. Dominguez-Morales, A. Jimenez-Fernandez, and G. Jimenez-Moreno, (preprint) arXiv:2206.04924, **2022**.
- [50] H. Cui, R. A. Andersen, *Neuron* **2007**, 56, 552.
- [51] H. Scherberger, R. A. Andersen, *J. Neurosci.* **2007**, 27, 2001.
- [52] S. Poulter, T. Hartley, C. Lever, *Curr. Biol.* **2018**, 28, R1023.
- [53] S. A. Lobov, A. I. Zharinov, V. A. Makarov, V. B. Kazantsev, *Sensors* **2021**, 21, 2678.