

A FLY-INSPIRED MUSHROOM BODIES MODEL FOR SENSORY-MOTOR CONTROL THROUGH SEQUENCE AND SUB-SEQUENCE LEARNING

Paolo Arena, Marco Calí, Luca Patané, Agnese Portera

*Dipartimento di Ingegneria Elettrica, Elettronica e Informatica, University of Catania, Viale A. Doria 6
Catania, 95100 Italy*

E-mail: parena@dieei.unict.it, lpatane@dieei.unict.it

National Institute of Biostructures and Biosystems (INBB), Viale delle Medaglie d'Oro 305, 00136 Rome, Italy

Roland Strauss

*Institut für Zoologie III (Neurobiologie), University of Mainz
Mainz, Germany*

E-mail: rstrauss@uni-mainz.de

Classification and sequence learning are relevant capabilities used by living beings to extract complex information from the environment for behavioral control. The insect world is full of examples where the presentation time of specific stimuli shapes the behavioral response. On the basis of previously developed neural models, inspired by *Drosophila melanogaster*, a new architecture for classification and sequence learning is here presented under the perspective of the Neural Reuse theory. Classification of relevant input stimuli is performed through resonant neurons, activated by the complex dynamics generated in a lattice of recurrent spiking neurons modelling the insect Mushroom Bodies neuropile. The network devoted to context formation is able to reconstruct the learned sequence and also to trace the subsequences present in the provided input. A sensitivity analysis to parameter variation and noise is reported. Experiments on a roving robot are reported to show the capabilities of the architecture used as a neural controller.

Keywords: Insect mushroom bodies; bio-inspired control; spiking neurons; learning; neural model; resonant neurons

1. Introduction

Bio-inspired robotics is nowadays strongly influenced by the new results in understanding the insect behavioral capabilities to develop autonomous robotic systems able to deal with complex dynamically changing environments^{14,53,77} and adaptive navigation control.⁶⁴ Among insect species the fruit fly is a perfect model organism due to the relatively low number of neurons in the central brain (about 10^5) organized in different neuropiles, involved in a multitude of interesting behaviors.³⁰ Mutant and wild-type fly behaviors can be compared, through focused experiments, using genetic tools based on the GAL4 technique.³⁵ These procedures allow to discriminate which neural centre and learning mechanisms are responsible for each behavioral response.^{21,72,85} In previous works^{7,16,18} the authors developed an insect brain

architecture, mainly realized with spiking networks,⁴² to model some relevant functions of Mushroom Bodies (MBs) and Central Complex (CX),^{15,63} important centres of the insect brain. From Neurobiology it is known that MBs are mainly responsible for olfactory learning,^{44,54,75} but recently they were discovered to handle also the adaptive termination of behaviors, multimodal integration and a multitude of other behavioral responses, like decision making and motor learning.^{36,45} In recent works, the basic functionalities of the fruit fly MBs computational model, related to olfactory learning, were extended to perform more complex tasks among which are attention, expectation and delayed match-to-sample.^{20,22,23}

MBs show anatomical and functional similarities with mammalian hippocampus and cerebellum,³⁷ the latter having a role in motor sequence learning.⁵⁵ Such

functionalities were modelled through hetero-associative memory.^{32,52} The role of time is also emphasised in the locusts olfactory processing:⁸⁸ the encoding of complex natural stimuli such as odours may involve the precise timing of neural activity. All these results were exploited in our model, where the olfactory system and neural circuits are modelled as dynamical systems able to produce a series of behaviors. The time is mapped into a space-distributed dynamic which can serve a multitude of concurrent behaviors, among which sequence learning and reproduction.

In *Drosophila melanogaster* (DM), there are different types of olfactory sensing neurons, whose collective dynamics contributes to encode the features (e.g. odorant components) of the source providing the stimuli. The Antennal Lobes (ALs) are the first neuropile encountered in the olfactory path and consist of glomeruli linked to olfactory receptors that transfer information, through Projection Neurons (PNs), to the protocerebral areas.⁸⁰ The connection with the large number of MBs cells allows a boost in dimensionality useful to improve the representation space.^{25,44} At the same time PNs are connected to the Lateral Horn (LH). In locusts LH inhibits, after a delay, the activity of the Kenyon Cells (KCs) neurons.⁶⁷ Therefore the KCs receive a sequence of excitatory and inhibitory waves from the PNs and LH respectively and are believed to communicate with one-another through axo-axonal synapses.²⁹ The distance between two consecutive waves defines the time window used for information processing through the maintained spiking activity in the KCs.⁴⁶ In the proposed model we considered a time window of $80ms$. These characteristics were used in literature to analyse pattern formation^{66,67} and to formulate non-elementary learning mechanisms.⁸⁹ When modelling complex neural activities to generate time-varying signals, two among the main strategies are either to use suitably controlled chaotic dynamics,^{8,73} or to explore Reservoir Computation, by extracting the needed dynamics using read-out maps.^{51,56} The key feature introduced in this paper is that, in order to address a number of different functionalities into such a tiny structure, like the MBs, it is necessary to migrate from a traditional, block-size computational model²⁸ (where each sub block is in charge for a particular defined function) to a holistic view of brain processing, mainly related to the concept of emergence and complex dynamics generation. This concept has much in common with the modern theory of Neural Reuse,⁴ which is here exploited for the emergence of different behaviors in parallel.

In this paper we modelled the KC layer as a dynamic spatial temporal pattern generator extracting the relevant dynamics needed to perform a behavioral task by training multiple read-out maps that mimic the role of extrinsic neurons in the biological system as described in details in recent works.^{24,25} After analysing interesting studies on Neural Dynamics model, inspired by the biological nervous system, and applied for problem solving,^{1-3,69} we focused our attention on different kinds of spiking-based networks.^{42,90} Finally we decided to model the KCs activity as a Liquid State Network (LSN), a lattice of connected spiking neurons similar to a Liquid State Machine,⁵⁶ that contains mainly local synaptic connections (as in a Cellular Nonlinear Network structure^{5,11} already used for locomotion control¹²), resembling axo-axonal communication among the KCs.²⁹ This lattice modulates sensory information, creating a dynamic map, which can be exploited concurrently both for classification and for motor learning purposes. In the former case, a signal of a specific, pre-assigned frequency is generated for each output class; in the latter case, by trial and error, the system can adapt the motor parameters so as to obtain a rewarding limb motion for the assigned task. The time-varying signals modulating those parameters, for instance to perform a climbing action (e.g. generated through a series of trials), can be extracted from the dynamics contained in the MB-lobes. KC neural circuitry is therefore re-used concurrently for a number of different aims, as in the line of the Neural Reuse, which has been addressed as a property particularly worth to be studied in the tiny insect brains.⁶⁵ Furthermore, in contrast with the previous model discussed in Refs. 22 and 6, in the architecture here formulated, the KCs dynamics does no longer converge to a cluster of activity directly related to the input. Taking into account also other approaches as reported in Ref. 39, we performed the classification task exploiting the sparse dynamics generated within the KC lattice of neurons implemented using a LSN, which now is not required to reach an equilibrium in the firing rate. Instead multiple read-out maps exploit this far-from-equilibrium dynamics (as found in Ref. 71) and are trained to stimulate resonant neurons representing specific classes. The role of resonant neurons in classification of auditory stimuli was already demonstrated in several works related to other insects like crickets^{10,74,87} and also for classification of mechanical data acquired by a bio-inspired antenna system.⁷⁰ This approach allowed to consider also motor learning capabilities, where learning the time evolution of reference signals is a key element,⁵ within the

MB computational model. Therefore we can both classify static features and learn time-dependent signals to be exploited as references for the motor system. Several attempts were performed to model the MBs for classification with structures mainly based on spiking neurons.^{61,76} An important difference with our model consists in the introduction of a new layer, named Context layer, needed to include sequence learning capabilities to the architecture. The KCs, when stimulated, produce sparse activity^{67,78} that has been associated to classes with reinforcement learning in a structure similar to a Support Vector Machine (see Refs. 48 and 47). From the other hand, our architecture exploits the complex internal dynamics that is extracted and condensed in periodic signals whose frequency is able to stimulate specific resonant neurons. We considered an unsupervised growing mechanism to create new classes and a supervised learning to train the read-out maps. To include sequence learning capabilities to the proposed system a Context layer was introduced. This structure is responsible for storing and retrieving sequences of events. Even if its position within the insect brain is not yet identified, the MB lobes are good candidates.⁸¹

The size of the proposed network in terms of number of neurons and connections was down-scaled with respect to the biological counterpart to reduce the computational time in view of a robotic implementation. The idea is to consider the Context layer as composed by groups of neurons topologically organized in concentric circles. We know that this kind of arrangement is present for instance in bees, where the calyx neuropile is concentrically organized.⁸⁴ Moreover, patterns of genetic expression in DM revealed that KC axons projecting into the γ lobe form the circumference of the peduncle, whereas a quartet of axon bundles form the core of the peduncle and project into the α and β lobes:³³ a concentric axon bundle is thus present in the fly MBs. In our model we hypothesize that each ring is stimulated during an input presentation and we assume that neural activity moves from the inner to the outer ring in time following a diffusion-like process. The Context neurons are connected to the resonant neurons where they compete with the current input information to produce the output of the network through non-elemental learning processes^{43,89} that are at the basis of the expectation and sequence learning schemes. In Ref. 58 a spiking network was trained using bio-inspired learning mechanisms to generate temporally ordered patterns of neuronal activity, similar to those ones observed in the mammalian cortex, in response to repeated se-

quences of sensory inputs. The network is divided in multiple layers and works as Winner-takes-all reinforcing synaptic connections among consecutive clusters. In our architecture we adopted resonant neurons for classification and we introduced a Context layer to learn complex sequences where the same input can be repeated multiple times in the sequence and to handle also sub-sequences that can be extracted from longer sequences if relevant for the on-going task. As far as sequence learning is concerned in *Drosophila melanogaster* (DM), the experiments reported in Refs. 62 and 57 represent a first attempt where the DM behavior was tested in a multiple T-maze scenario. The results suggest that, in some configurations, DM statistically prefers to persist in the selected chosen action (e.g. left turn) repeating it multiple times (sequential repetition) whereas in a different set-up it prefers to switch the decision in time (sequential alternation). Therefore, to the best of our knowledge, the presence of sequence learning in flies is up to now unclear. Our effort is also in this direction: we are not arguing that DM is able to show sequence learning, but that in principle its brain contains the neural circuits and learning functions needed. The obtained simulation results can be the starting point for new biological experiments to answer to this open question. To support our hypothesis the ability of DM to face a Morris water maze-like experiment seemed to be out of DM behaviors until recently demonstrated.³⁸ The developed computational model was used as a neural controller for a roving robot facing with different scenarios inspired to experiments performed with honeybees⁹¹ that are able to use visual features of objects to make a series of decisions while negotiating a complex maze and learn rules that allow to fulfil the assigned tasks. The basic idea is to consider each element of a sequence as an external input coming from the sensory system. The sequence is treated as general as possible: each element could represent either an odour, a visual pattern, or another external, or even endogenously generated input. After the input processing the system performs an action/behavior associated to that input (e.g. a following action) and in the meantime associates to the stimulus the corresponding motor action performed (e.g. left turn and forward movement). When, during the recall phase, the input is noisy or absent, the neural network evokes the learned sequence looking for the next expected element and performs the associated series of actions.

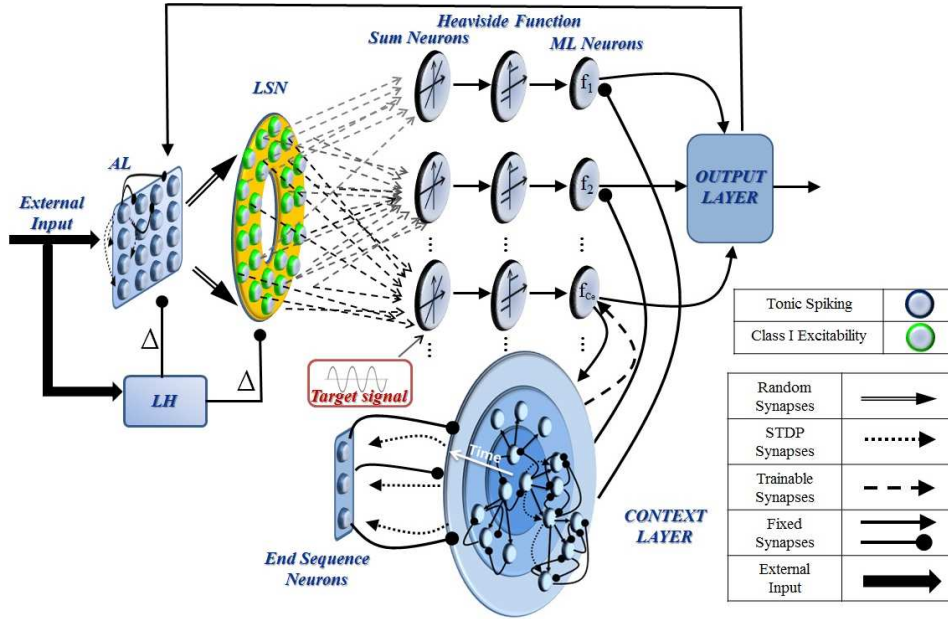


Figure 1. Block scheme of the developed architecture (see text for details).

2. MB-inspired computational model

The computational model inspired by the MBs neuropile processes external inputs, acquired in terms of general features of the relevant objects present in the scene, using the ALs. This neural assembly is realized through a 4×4 network of spiking neurons interconnected with learnable synapses, used to filter out noise and to reconstruct missing features.²⁰ The neural activity of the ALs is then projected, through random connections with probability of 25%, to the LSN. The LH is activated by the external input and, after a delay (i.e. 80 ms in the following simulations), it inhibits both the ALs and the LSN resetting the neural activity and allowing the system able to process new sensorial stimuli. Each neuron within the LSN is connected, through trainable synapses, to neurons acting as weighted integrators (here called Sum Neurons) creating a read-out map of the LSN activity. As outlined above, in line with the theory of Neural Reuse, the structure designed can elicit concurrently different behaviors for different purposes depending on the structure of the read out map. Some behaviors, already presented in other contributions,^{5,20,23} will be briefly discussed in Section 4. Here the classification and the sequence and sub-sequence learning, and later on, decision making processes, are described in detail. With regards to classification, each class is represented by a resonant neuron realized through a Morris-Lecar model.⁶⁰ A training procedure, described in the next section, allows to as-

sociate to each different input a resonant neuron representing a new class. The read-out map is trained in a supervised way: a periodic wave has been chosen as target signal with a frequency able to stimulate the corresponding resonant neuron. The first read-out map is trained to generate a sine wave with a frequency of 62.5Hz when the first input signal is provided. When resonant neurons are not excited by the input a new read-out map is generated and a new frequency is associated. The frequency range here used spreads from 50Hz to 250Hz. This interval was chosen to allow the coexistence of about five different classes. Lower frequencies are infeasible because at least five periods are needed to have a reliable number of spikes in the resonators for a robust classification. To learn a higher number of classes, the time window defined for the target signal (i.e. 1000 samples with a sampling time $dt=0.08ms$) should be increased to host enough samples. Moreover in this case the time constants used in the LSN should be tuned to generate the frequencies needed. A minimum number of spikes (larger than 50% of the maximum allowable depending on the frequency) has been fixed to consider the resonator in an active state. The signal coming from the Sum Neuron is shaped by a Heaviside function, before entering into the resonator, to filter-out noise. As depicted in Fig. 1, the Context layer structure is organized in concentric rings (only three rings were reported for the sake of simplicity). The first ring contains a number of neurons equal to the current num-

ber of classes (N_c). In the second ring, for each neuron of the first ring, there are N_c neurons, and so on. This implies that, within each ring, there are $N_c^{N_r}$ neurons, N_r being the ring number. The potentially large number of neurons building-up the Context layer is justified because we are simultaneously learning both sequences and subsequences. This possibility boosts the capabilities of the structure, much beyond the simple sequence learning. The number of rings considered defines the maximum sequence length. Lateral inhibition among neurons of the same ring generates a competition, filtering out potential disturbances.⁴³

Sequence learning takes place through different sequential stages here called epochs characterized by a neural activity either stimulated by an external input or an internally generated one to recall a missing element already stored in the network. During each epoch information propagates one ring ahead, from the inner to the outermost. So rings represent time steps characterising the sequence length. Every epoch, the winner neuron in the outermost ring visited for this epoch and the winner neuron of the previous ring are subject to a learning process which modulates their connection weights using Spike Timing Dependent Plasticity (STDP),⁷⁹ as performed in several related works.^{9, 17, 68, 83} In our model, this process could cover larger time scales than the standard STDP. This is required to create correlations among consecutive objects whose presentation does not happen within the usual STDP time window. This aspect has been also theoretically treated in Ref. 39 where specific spike-based plasticity rules were proposed. Furthermore biological evidences of this phenomena are reported in literature for insects and specifically on DM⁸⁶ and can be modelled using memory traces and reverberation mechanisms.^{34, 50}

Each neuron in the Context layer is also linked to an End Sequence Neuron with STDP synapses. In details all neurons in each ring are connected to the corresponding End Sequence Neuron within an End Sequence Vector (whose length matches the number of rings in the Context). A rewarding signal, following the completion of a sequence, activates the End Sequence Neuron for the outermost ring of the Context layer. The synapse connecting this End Sequence Neuron with the winner neuron in the outermost ring will be thus reinforced. In our model the reward signal acts as a dopaminergic stimulus on the End Sequence Neuron to reward the sequence just completed and reset the activity in the Context layer to start a new sequence.⁵⁰

2.1. The Liquid State Network

The LSN is composed by excitatory (75%) and inhibitory (25%) neurons. The synaptic weight values are randomly distributed between -0.5 and 0.5 whereas the input weights are fixed to 1. The generation of the synaptic connections within the lattice is based on a probability that is a function of the distance $d_{i,j}$ between the presynaptic (i) and postsynaptic (j) neurons.

$$P_{ij} = k * C_{i,j} \quad (1)$$

where $C_{inh,inh} = 0.2$, $C_{inh,exc} = 0.8$, $C_{exc,inh} = 0.4$, $C_{exc,exc} = 0.6$. and

$$\begin{aligned} k &= 1 \quad \text{if } d_{i,j} \leq 1 \\ k &= 0.5 \quad \text{if } 1 < d_{i,j} \leq 2 \\ k &= 0 \quad \text{if } d_{i,j} > 2 \end{aligned} \quad (2)$$

The parameters $C_{i,j}$ have been chosen according to Ref. 56. The distance is calculated considering the neurons distributed on a regular grid with toroidal boundary conditions. The distance $d_{i,j} = 1$ is considered for both horizontally and vertically adjacent neurons. The time constant of the synaptic model was randomly chosen among the values $\tau = 2.5, 5, 15$ and $25ms$. This variability improves the dynamics that can be generated inside the network within the processing time window. The Sum neurons, modelled with a linear transfer function, are massively connected with the LSN. The weights of the readout map are randomly initialized around zero and are subject to learning.

2.2. Neural models and learning mechanisms

Different neuron models were used in the architecture to generate the suitable dynamics needed in the subsystems: the Izhikevich spiking neurons proposed in Ref. 49 and widely used to model brain regions,⁸² and the Morris-Lecar model (ML) in Ref. 60. As indicated in Fig. 1, we adopted the Izhikevich Tonic spiking model in the ALs, Context layer and End Sequence neurons whereas the Class I model was exploited in the LSN. Classification is performed using the ML neurons that can emit spikes only if stimulated with a periodic input showing a particular frequency; the neuron characteristic frequency can be tuned acting on the model parameters.⁷ The synaptic model transforms the spiking dynamics of the pre-synaptic neuron into a current that excites the post-synaptic one.

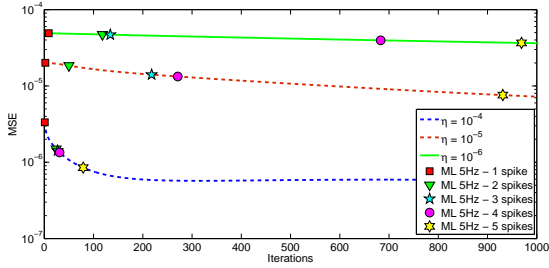


Figure 2. Trend of the mean square error during the iterative process performed to train a read-out map associated with the resonant neuron at 62.5Hz. The markers indicate when the neuron is adequately stimulated in order to emit a certain amount of spikes. By changing the learning rate from 10^{-5} to 10^{-3} we can speed up the process, although, using $\eta = 10^{-3}$ can cause oscillations in the error evolution. The level of MSE obtained using the pseudo-inverse method is about 3.5×10^{-7}

Different learning strategies are used in the proposed MB architecture to generate the desired behavioral responses. In particular the synaptic weights are modulated through experience using STDP.

A decay rate allows the system to work in dynamically changing environments: here the learned sequences could be forgotten if no longer rewarding. Interesting applications of this learning paradigm to biorobotics, together with details on the parameters, are reported in Ref. 13. The role of the connections between the Context layer and the ML neurons is to strongly inhibit all the ML neurons except that one representing the next expected class C_e whose synaptic weight is subject to learning during the sequence presentation. A Hebbian learning process is used to update the synaptic weights between the Context layer and the ML neurons when pre- and post-synaptic neurons are concurrently active. When such synapses, after learning, are sufficiently strengthened, the ML neuron corresponding to C_e will fire, contributing to stimulate C_e in the next ring of the Context. Finally spiking neurons in the KC lattice are fully connected to the Sum Neurons via plastic trainable synapses. A simple supervised learning method based on the pseudo-inverse algorithm⁵¹ has been adopted as well as an incremental learning rule as also illustrated in Ref. 59 where the spiking activity of the neurons is transformed in continuous signals using different kernel functions allowing the evaluation of an error needed for the learning. Different supervised learning methods, based on back propagation for spiking networks,^{40,41} have not been adopted due to the presence of recurrent connections at the level of the LSN. Although different learning approaches (e.g. based

on minimizing an upper bound on the KullbackLeibler divergence from the target distribution to the model distribution) could be adopted,²⁷ we decided to introduce a simple incremental learning strategy based on the Least Mean Square algorithm, that adapts the synaptic weight on the basis of the computed error and the local activity generated by the pre-synaptic neuron, working with the synaptic response instead of directly with the spike train.

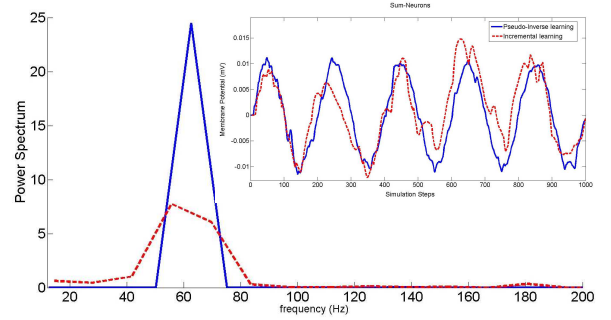


Figure 3. Comparison between the results of the read-out map obtained using a pseudo-inverse (solid line) and an incremental learning process (dashed line). The output of the Sum neuron and the Power spectrum of the signals are reported.

For a given Sum Neuron s , the weight value at each integration step depends on the lattice activity and on the error between the current output and the desired target value. This can be summarized in the following equation:

$$W_{i,j}^s(t+1) = W_{i,j}^s(t) + \eta * Z_{i,j}(t) * E^s(t) \quad (3)$$

where η is the learning factor, $Z_{i,j}(t)$ is the synaptic output of the neuron (i, j) of the lattice at time t and $E^s(t)$ is the error between the desired target and the Sum Neuron s . Similar results could be obtained cumulating the error in each presented sample and updating the weights at the end of the presentation. Moreover to stabilize the learning process an error threshold ($E_{th} = 10^{-8}$) is imposed to avoid a negligible weight update for each sample presentation. The trend of the mean square error for the incremental learning process is shown in Fig. 2. The exit condition for the learning process consists in monitoring the spikes emitted by the ML neurons and, when a given number of spikes is reached, learning is stopped and the read-out map stored in the architecture. To evaluate the performance of the learning process, a comparison between the pseudo-inverse (standard solution) and the step-by-step method, more biologically plausible, was performed. The results, shown in Fig. 3, illustrate that using an incremental procedure the learning process converges to a signal able to properly stimulate the corre-

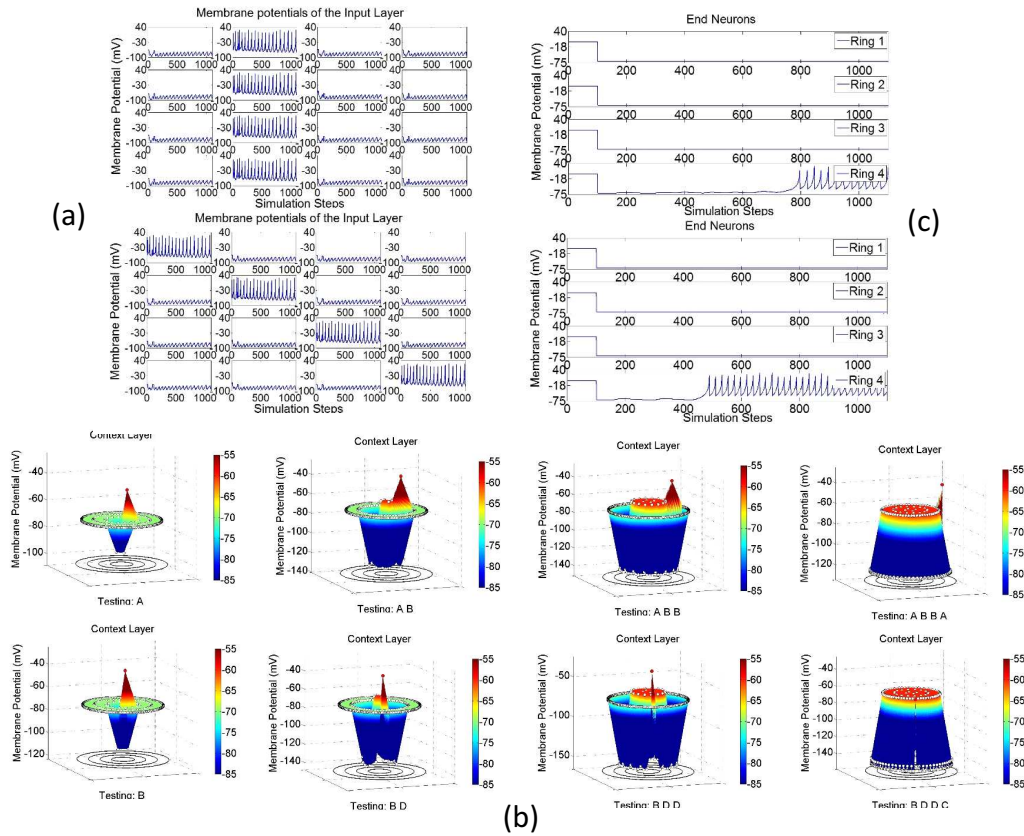


Figure 4. Internal simulation of the system while recalling the sequence *ABBA* and *BDDC* in the first and second row of each subfigure respectively. The AL input layer for the first element of each sequence is reported (a) and followed by the Context activity internally generated by the system (b) until one of the End Sequence Neurons is excited (c). The number of emitted spikes is used to select the most rewarding sequence that in this case is *BDDC*. The empty circles indicate the neurons enrolled in each ring of the Context layer, the most active (i.e. winner) neuron is indicated with a filled circle.

sponding ML neuron by emitting that maximum amount of allowable spikes. However, the output trend of the Sum neuron does not fully match the sinusoidal target.

In the simulations reported in the following sections, the pseudo-inverse method was adopted to speed-up the learning process.

3. Simulation results

To analyse the network activity, a simple sequence of three different elements *ABC* was provided to the system. The presented objects are identified through a set of features that excite the AL layer. Here and in the following simulations the objects are discriminated using four different features, each one characterized by four different values. The AL layer is directly linked through random connections with the KC network, characterized by a sparse spiking activity. When the first input is provided, due to the absence of already established classes, a new

read-out map is created and trained to generate a periodic signal able to stimulate the associated ML neuron created to resonate at 62.5Hz . Between the Sum neuron and the ML neuron a Heaviside function is introduced to avoid classification problems, normalizing the amplitude of the stimulus and creating an impulse train. The active class neuron is linked to the Context layer that now contains a single neuron placed in the inner circle. When the second input is presented, the ML neuron previously created is not correctly excited and a new read-out map is generated and associated with a new ML neuron tuned to resonate at a different frequency. At the same time new neurons are enrolled in the Context layer because a new class has been created and connections between these neurons and the ML neurons are established. The Context neuron of the first ring diffuses in the second ring and the interaction with the active class neuron determines the new winner in the Context layer. Finally to reinforce the link between

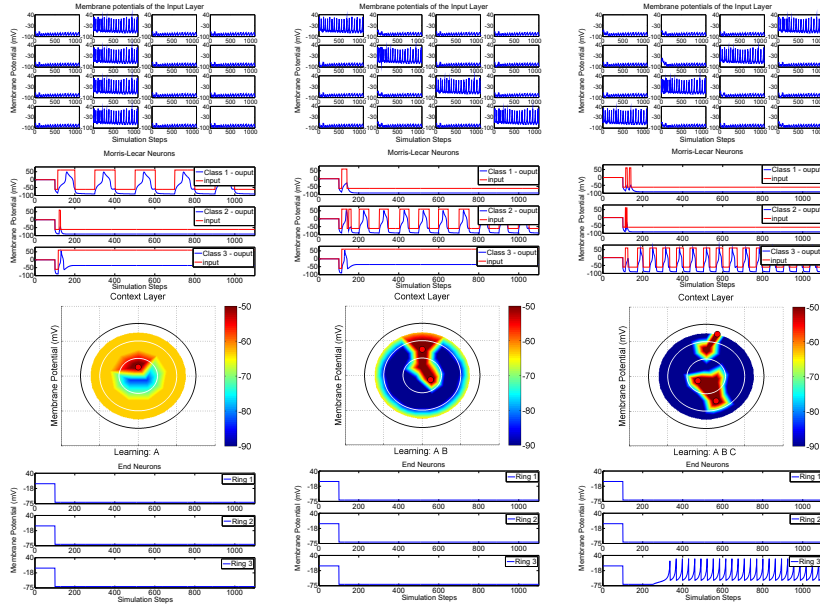


Figure 5. Activity of the network when the sequence ABC is provided and also subsequences are retained. Each row represents the activity of the AL layer, ML neurons, Context layer and End Sequence neurons respectively. The filled circles represent the active neurons: multiple chains are generated to trace the subsequences BC and C in the Context layer.

the previous element of the Context and the actual active resonant neuron, their synaptic connections are strengthened. The same process is repeated for the third input that is associated with a new class neuron. In this case a rewarding signal is provided, the End Sequence neuron associated with the third ring of the Context layer is excited and the connection between the active Context neuron in the outer ring and the End neuron is reinforced through STDP. The End Neuron resets the Context activity and the same sequence is represented multiple times to improve the strength of the learned synapses that contribute to the memory formation. For the parameters chosen, about 4-5 presentations are enough to create a well established memory. In the output layer (see Fig. 1) the activity of the so-called Persistence and Sameness neurons, already modelled in previous works,²⁰ is concurrently stimulated and can interact with the learning process. When the objects are presented for the first time to the network there is no activity in the output layer: the subsequent presentations generate an activity that can be used both to identify the repetition of the same input as in the delayed matching-to-sample task (i.e. Sameness neuron) and to polarize the attention to a certain stimulus if continuously presented.²⁰

3.1. Decision making process

The architecture allows to store and consequently retrieves multiple sequences to perform a decision making process. A rewarding signal, provided at the end of each sequence, acts as a quality indicator. In our model the number of spikes of the End neuron indicates the level of reward associated with the generated sequence. As an example in Fig. 4 after learning both sequences $ABBA$ and $BDDC$, the system is requested to choose which sequence to generate in front of the presentation of the first two objects of the two sequences (A and B). This is a typical case of landmark sequence following to reach a feeding place. The system internally simulates one and then the other sequence and compares the number of spikes of the End Sequence neuron in both cases. In the reported example the sequence $BDDC$ is the most rewarding one and then the behaviors related to this sequence are performed.

3.2. Learning sequences and subsequences

Further exploitation of the possibilities offered by the architectures consists in learning not only the sequence of the presented elements but also the subsequences included therein. Here we are widening the potentialities offered by the model, which is still based on MBs, but improved to provide interesting properties useful for real

applications. This is obtained allowing each presented element to activate, besides a neuron in the ring of the Context Layer standing for the current time step in the sequence, also a neuron in the first ring of the Context.

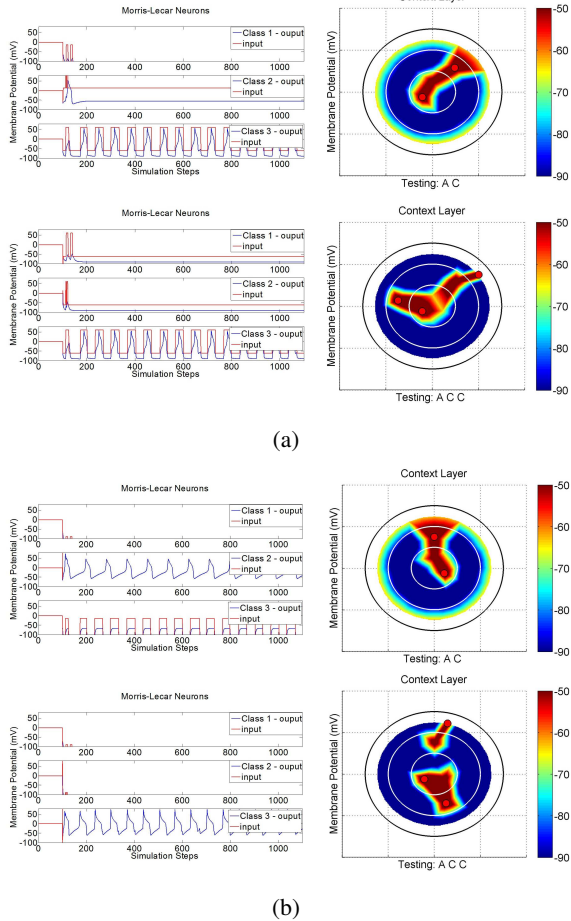


Figure 6. Behaviour of the ML neurons and Context layer during the presentation of the second and third element of the sequence ACC when the sequence ABC is not completely learned (a) and when the memory trace is well established after multiple presentations (b). In (a) the fresh information provided in input wins the internal competition whereas in (b) the Context activity prevails over the external input, treated as disturbance in order to reach the final reward.

In this way, independently of which is the first element of the sequence, each element is treated as a starting point. This leads to generate and store multiple chains of context activity: in presence of very noisy and complex conditions this strategy could be useful to retain only statistically relevant sub sequences. In fact, depending on how frequently the sub sequence is presented, this can be reinforced more than others and considered as most reliable to be generated when the initial element is presented.

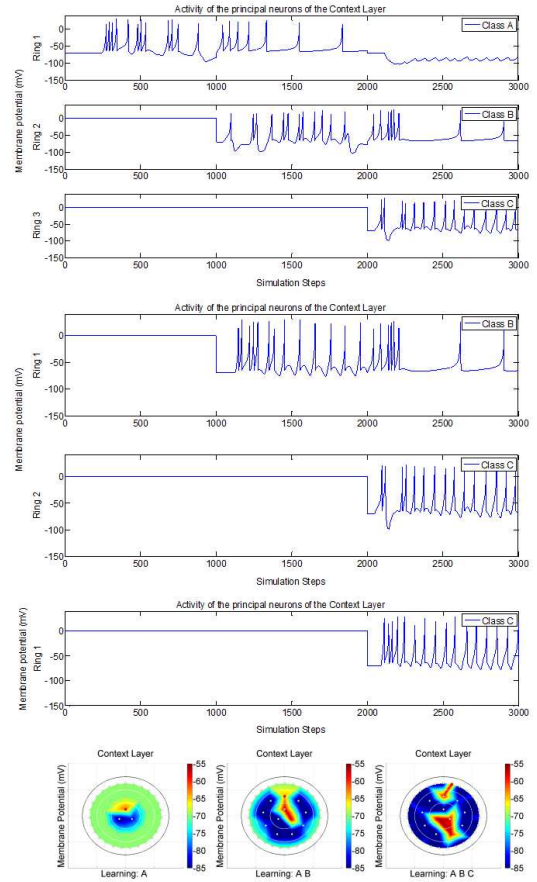


Figure 7. Behaviour of the active neurons in the Context layer when the sequence ABC is presented. During the first 1000 simulation steps the presentation of A sustains the activity of the corresponding neuron (Class A) in the first ring; afterwards the second element B activates the associated neuron both in the second and in the first ring starting a new subsequence. In the meantime the activity due to object A is still present in the first ring even if with a slight decrease in time: this allows STDP learning that will reinforce the synapse connecting the neuron A in the first ring and neuron B in the second ring. The process continues also for the object C . Circles indicate neurons, the winner is filled and the colour map indicates the mean membrane potential activity in the Context layer.

An example of learning subsequences is shown in Fig. 5, where the evolution of the Context layer while learning the sequence ABC is reported. Each element of the sequence creates a trace in the Context layer that starts from the inner ring and propagates towards the outer ones. The memory trace in the Context layer is incrementally strengthened through multiple presentations of the same sequence. Normally the external input guides the network activity as shown in Fig. 6 (a) where, after a few presentations of the sequence ABC , a new sequence

ACC is given. From the other hand, as shown in Fig. 6 (b), if the rewarding sequence is repetitively presented (about 4-5 presentations are needed for the actual parameter choice), the information stored in the Context layer in terms of synaptic weights, becomes more relevant to win the competition with external inputs, considered as disturbances to be filtered out, in order to obtain the expected final reward.

As previously introduced, the learning between the Context layer and the ML neurons, as well as the learning among the rings of the Context layer, is performed between two processing windows related to two consecutive stimuli presented at different times. In view of a robotic implementation, the time between two consecutive simulation windows was not taken into account. Basically our idea was that the actions performed by the robot are relevant for the next object presentation. For instance in a multiple T-maze experiment a variable time interval can pass when the robot, after executing a behavior, is involved in looking for other information in the scene (see Section 5 for more details). The time lapsed between two sensorial acquisitions could therefore be of the order of seconds. Even if STDP works in time scales in the range of tens of milliseconds, learning processes can be expanded: a plausible solution is proposed by using a neuron reverberation mechanism.³⁴ The neural activity, when the input stimulus is switched off, persists in time with a continuous decrement of the firing rate. In this way a trace of the residual activity can persist and this causes STDP adaptation even after seconds. In our model, an example of the reverberation phenomenon in the Context layer is reported in Fig. 7 where the time evolution of all the active neurons in the Context layer is shown during the presentation of the sequence *ABC*. Learning of multiple subsequences is evident from the contemporary activation of the actually presented element in each active ring. The neural activity after the presentation of the stimulus slowly fades, maintaining a residual spiking activity within the successive simulation window and allowing STDP with other Context neurons. When the simulation window of 1000 steps (with a $dt=0.08ms$ for a total of 80ms) is completed and the input current for the Context neuron, generated by the ML neurons is finished, in the successive simulation window a residual current is provided. Therefore the previously active neurons prolong their activity in a fading way. The input is fixed to 10 pA that represents a typical mean input value when the ML neurons are active and decreases with a time constant $\tau = 400ms$. In this way the spiking rate decreases until

the input current is not enough to stimulate the neuron.

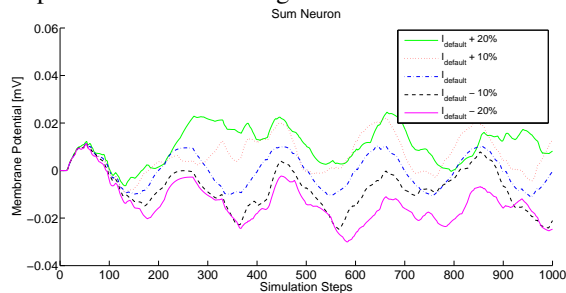


Figure 8. Sensitivity to the input current when the fourth feature of object A is affected by noise after the read-out map learning.

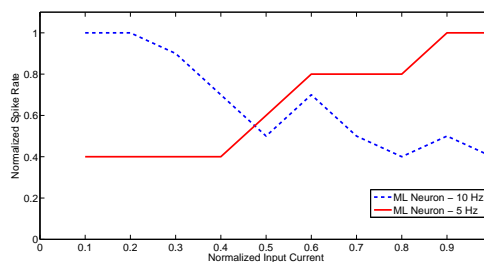


Figure 9. Normalized activity of the two ML neurons associated with the class A (5Hz ML) and E (10Hz ML) when the input neuron associated to the last feature is increasingly activated from a completely missing information (value 0) to the full presence (value 1).

3.3. Sensitivity analysis

To better understand the robustness of the system to noisy signals and parameter variations a sensitivity analysis was performed. In particular we started by analysing the behavior of the LSN when the input current of one feature is modified with respect to the default value used during learning. In Fig. 8 an example is shown of how the learned sinusoidal waveform at $62.5Hz$ is affected by noise applied to the current associated with one of the features coming into the AL input layer. The behavior is robust for a wide range of perturbations, until $\pm 10\%$ if a single feature is affected and about $\pm 5\%$ if all features are subject to noise. Beyond these values the activity of the sum neuron will no longer be able to trigger the resonator. An interesting analysis was performed in relation to the creation of basins of attraction between similar classes. In particular we considered the learning of two different classes, *A* and *E* that differ only for the value of the last feature that in *E* is missing, therefore no input neuron is stimulated referring to the missing feature. During the learning phase pattern *A* was provided with noise of 5%

and 10% that reduced the value of the last feature and pattern E with noise that increases of the same quantity the last feature otherwise not present. During the testing phase, the response to the complete range of values was analysed and the normalized activity of the two ML resonant neurons, associated with the two classes, is shown in Fig. 9.

Table 1. Robustness of the network behaviors when the synaptic weights involving the Context layer are modified for two distinct cases: ($W_{ML-to-C}$ vs W_{exc-C}) and (W_{inh-C} vs $W_{ML-to-C}$).

$W_{ML-to-C}$	(a) W_{exc-C} (b) W_{inh-C}	Ratio	Beh.(a)	Beh.(b)
3.5	3.5	1	✓	✓
7	7	1	✓*	✓*
1.75	1.75	1	P_{1a}	P_{1b}
7	3.5	2	✓*	P_{2b}
3.5	1.75	2	✓	P_{2b}
10.5	3.5	3	✓*	✓*
3.5	1.17	3	P_{2a}	P_{2b}
1.75	3.5	0.5	P_{1a}	P_{3b}
3.5	7	0.5	P_{1a}	P_{3b}
3.5	10.5	0.33	P_{2a}	✓**

A smooth transition between the two classes can be appreciated, as expected in a reliable classifier when representative input patterns are next one another for the difference of only one feature. Another important aspect investigated refers to the robustness of the network to parameter variations. Due to the presence of multiple parameters, we considered the most salient elements of the Context layer that could be crucial for memory formation. We started considering the synaptic connections from the ML neurons to the Context layer ($W_{ML-to-C}$) and a comparative analysis with the excitatory synapses within the Context (W_{exc-C}) was performed. Table 1 reports the behavior of the system when the absolute value and the ratio between the synaptic weights of these fixed synapses is changed. The network is robust and correctly works with different weight configurations (symbol ✓) besides the default values reported in the first row. In some cases, the behavior is still correct but the maximum spiking rate increases creating potential problems in extracting the correct winning neuron (symbol ✓*). Problems occur when either the weight value is too low to correctly stimulate the post-synaptic neurons or the ratio is significantly modified. In particular two specific problems were identified: P_{1a} - due to the poor activity there is a wrong selection of the winner neuron in the Context

Layer and the correlation between the Context Layer and the End sequence neurons is not correctly learned; P_{2a} - the unbalancing due either to the strong excitation coming from the previous ring in the Context or due to the predominant contribution coming from the ML neurons produces an incorrect selection of the winning neuron.

A similar analysis was performed taking into account the inhibitory synapses in the Context layer (W_{inh-C}) instead of the excitatory ones as also reported in Table 1. All the network functionalities are maintained for many weight configurations. The problems identified are: P_{1b} - due to the reduced activity caused by low values for the $W_{ML-to-C}$, there is a wrong selection of the winning neuron in the Context Layer, in fact the reverberation activity becomes relevant maintaining the previously selected winner; P_{2b} - the effect of the inhibition is too low and there is no longer a unique winner among the neurons of the same ring in the Context Layer; P_{3b} - when the inhibitory effect is too strong, the network activity in the Context Layer disappears and it is not possible to find a winning neuron. Moreover, in some cases, the behavior is still correct but the inhibition is excessively strong producing very low values for the membrane potential (symbol ✓**).

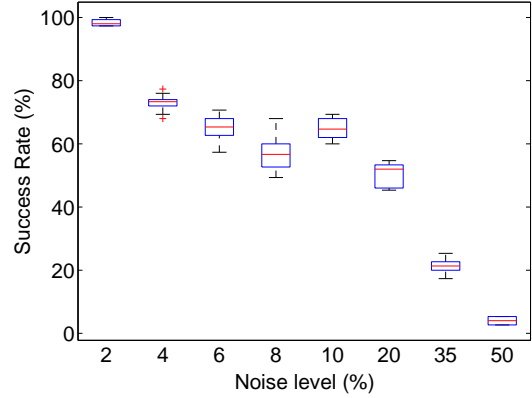


Figure 10. Success Rate, in terms of sequence learning capability, as a function of the added noise level (see text for details).

Concerning the effect of noise in the system, the robustness of the network was evaluated introducing white noise in the input current of each neuron, both in the external input and in input currents coming by the synaptic connections with other neurons. All the simulations were performed adding noise that dynamically changes each integration step, in a range from $\pm 2\%$ to $\pm 50\%$. The obtained results, depicted in Fig.10 show that, as the noise level increases the success rate decreases since an incor-

rect neuron can be selected as winner in the Context layer causing a different sequence to be learned or reproduced.

4. Remarks

The model presented has a number of interesting features. The signals generated by the Sum Neurons, thanks to the dynamics embedded into the LSN, can be used as pattern generators. Besides classification and sequence learning, as discussed in this paper, the structure is suitable also for motor learning, providing time dependent signals to control parameters related to the locomotion system. The same LSN can be so utilised and the motor learning feature can be included simply by adding another set of read-out maps. The peculiarities of the strategy for motor learning can be directly adopted from Refs. 5. In this way, the model built and the simulation results are a typical example of implementation of the concept of Neural Reuse. Moreover, in the current model, due to the small size of the LSN (8×8 neurons), massive connectivity between this lattice and the Sum Neurons is assumed. This could be relaxed in larger sizes, using a randomly based connectivity. Another characteristic worth to be discussed regards the target signals for classification: these, in the current model are generated, for the sake of simplicity, via a sinusoidal signal.

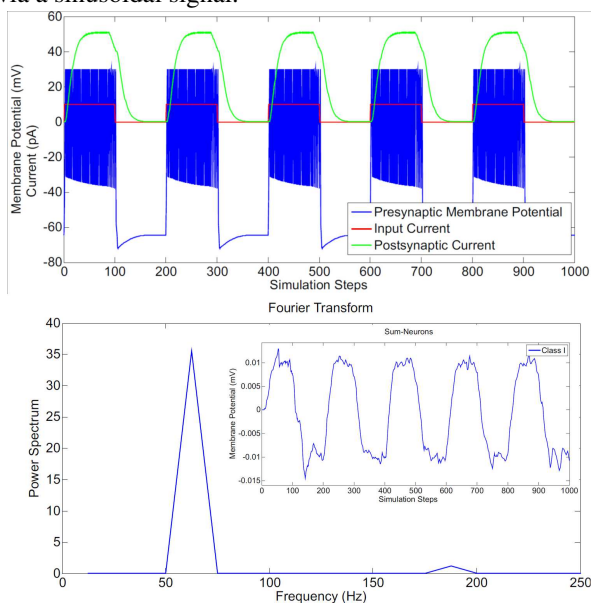


Figure 11. Normalized activity of the bursting neuron is shown in the first panel when the input current is a square waveform with amplitude $A = 10pA$, period of $62.5Hz$ and duty cycle 50%. The time evolution of the post-synaptic current is also reported. The learned signal reproduced by the Sum neuron and its power spectrum are reported in the second panel.

However, a more biologically relevant target signal can be provided using a target neuron. This can be assumed to deliver to the MBs information coming from general sensory systems or other parts of the brain, to create a match between the information produced by the MBs and that (e.g. sensory) signal. This process has deeply been studied in MBs. There are a lot of extrinsic neurons that innervate MBs, constituting input and output stations among these centres and a lot of other brain areas. To make our model next to this representation, instead of a simple sinusoidal wave, a more biologically plausible burst of activity was considered. An Izhikevich Phasic bursting neuron can be used to generate a suitable train of bursts that are processed via a synapse, and the post synaptic current is then used as a target signal for the read-out map learning.⁴⁹ In Fig. 11 the normalized activity of the bursting neuron is provided when the input current is a square waveform with amplitude $A = 10pA$, period of $62.5Hz$ and duty cycle 50%. The results of the learning process are also reported together with the Power spectrum of the learned signal where the predominance of the $62.5Hz$ component, needed to stimulate the resonator, is evident. Another aspect regards the output neurons: these act as integrators with different time constants that can be tuned, depending on the behavior to be boosted. For example, the time constant fixes the upper limit for the number of repetitions of the same stimulus that the system can retain as single objects within a complex sequence. Beyond this number, the persistence behavior is elicited,⁶ discarding the input information to continue the on-going behavior.

5. Robotic experiments

Considering that honeybees are able to negotiate a maze by using symbolic cues,^{31,92} and ants are able to navigate through routes,²⁶ we embedded our control architecture on a roving robot to face with scenarios where the available paths from the entrance to the exit have to be learned.⁹¹ In Ref. 26 the capabilities of ants to learn routes on the basis of familiar scenes is modelled through an Artificial Neural Network that, using a whole panoramic image as input can learn the followed routes. Instead the proposed architecture extracts from the visual input the landmarks of interest and process the acquired landmark features using a spiking neural network to learn multiple sequences and the corresponding expected rewards. The robot is a differential drive and includes a PC on board and a series of micro-controller-based boards to handle the motors. The platform is equipped with two ul-

trasonic sensors for obstacle detection and a panoramic camera to identify the presence of landmarks in the environment. In the proposed experiments two monitors placed at the boundaries of the arena are used to dynamically show landmarks depending on the position of the robot in the maze. The visual input from the camera is processed using OpenCV libraries for segmentation to identify landmarks and their features to be used as inputs for the control network. The computational time to process the signals corresponds to a few seconds in the actual robot setup, depending on the data logging. Depending on the obtained results an action/behavior is performed: it consists of a turning action and a forward motion to reach a new branch of the maze. The two monitors allow to emulate the presence of landmarks in each branch of the multiple T-maze. To guide the robot behavior we used a double reversed T to indicate a right turn and a double circle for a left turn. During learning the correct actions to be performed are provided showing only one landmark each time in the monitor positioned in the correct turning direction (see Fig. 12 (a)). During the testing phase only one monitor is used and the structure of the maze is modified to demonstrate that the robot is able to solve it using the knowledge acquired in the previous phase (Fig. 12 (b)). The connections with the neurons associated to the correct behavior are reinforced during the learning phase and the presented objects stimulate the correct movement to fulfil the task.

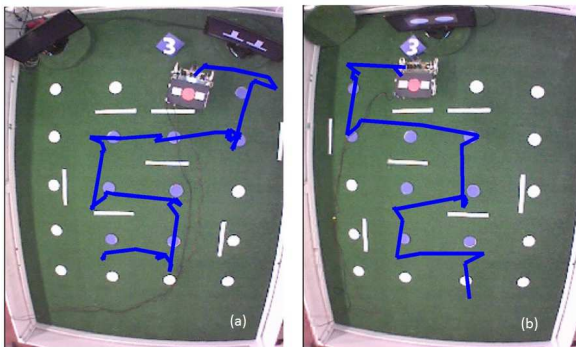


Figure 12. Robotic experiments where different maze configurations are used during the Learning (a) and the Testing (b) phase. The solid line represents the trajectory followed by the robot. The scenario is equivalent to the biological experiments performed with honeybees.⁹¹ During the Learning phase the robot associates a turning direction to every element presented. In the Testing phase (b) the robot can solve a different maze using the landmark projected in a unique monitor.

These results show that the robot is able to negotiate a maze by using symbolic cues as shown by honey-

bees.⁹¹ We then included the sequence learning capabilities considering a more complex scenario. The robot initially learns two sequences of actions to be performed to solve the maze following two different routes that guarantee, at the end, reward signals with different values. The first sequence (i.e. *Inverted-T*, *Circle*, *Inverted-T*) is associated to the left, right and then left maze-branch selection, whereas the second learned sequence (i.e. *T*, *inverted-T*) is associated to a right and then left turning action. The reward level that modulates the stimulation of the End Sequence neuron, is lower for the first sequence than for the second one that should be preferred. During the testing phase the robot is placed in front of two concurrent stimuli (i.e. *T* and *inverted-T*) to perform a decision making process. In this way the architecture can internally retrieve different sequences, depending on the initial stimulus, and selects the most performing one depending on the spiking activity of the End Neuron, that encodes the cumulative level of rewards obtained, for that sequence, during the learning process. The control structure internally simulates the outcome of the two possible sequences to be followed and, on the basis of the expected reward, selects the most rewarding one (Fig. 13). Multimedia materials on the performed experiments are available on the Web.¹⁹



Figure 13. After learning two sequences, the robot selects the most rewarding one performing a decision making process. The learned sequences are depicted on the two sides; the solid line represents the selected trajectory during the testing phase.

6. Conclusions

The insect Mushroom Bodies were studied here to inspire the implementation of neural models able to deal with classification, sequence and subsequence learning. The model being able to concurrently reproduce also other features like persistence, expectation and motor learning, constitutes a clear example of Neural Reuse. The idea is to exploit the complex dynamics present within the MBs, here in part modelled using a LSN, through extrinsic neurons working as read-out maps. The key points that

guided the new design process, compared with the previously developed structures, include: the possibility to elicit completely different behaviors concurrently (classification, sequence and subsequence learning, persistence, expectation and motor learning), according to the paradigm of Neural Reuse; the presence of sparse spatial activity inside the LSN instead of cluster formation; the introduction of a growing mechanism for classification and the inclusion of multiple ways to handle the internal dynamics using several read-out maps mimicking the role of extrinsic neurons. Moreover the presence of a Context layer allows learning and retrieving of multiple sequences, including the possibility to extract subsequences from the whole stream of events. The learning process in the Context is based on STDP that allows, together with the reverberation mechanism, to correlate events with a large time scale. Decision making processes can also be reproduced considering the End Sequence stimuli as rewarding signals that can be modulated. In this way the architecture can internally retrieve different sequences, based on the initial stimuli. Then it can select the most performing one depending on the spiking activity of the End Neuron, that contains information about the cumulative level of reward obtained, for that sequence, during the learning process. The idea is to extract information from the performed simulations, to propose new experimental setup in order to validate the model and the hypotheses included. This step will lead to further tune the parameters according to the biological experimental results or to those coming from the literature. Our model is therefore the result of known facts in insect neurodynamics and of some hypotheses on their role in the processing of time related events: *Drosophila melanogaster* brain has all the ingredients to solve the sequence learning problem, even if, either for the lack of suitable experiments, or for the missing development of suitable read-out maps in its brain, such capabilities have not yet been discovered or shown. In one or the other case, we can state that MBs are a perfect candidate to be involved in this spatio-temporal learning process, representing a clear example of Neural Reuse.

Acknowledgement

This work was partially supported by the FIR 2014 Project *Self-reconfigurable, flexible, multi-body, lowcost, modular robot systems for exploration of dynamical, hostile or unknown environments* and by MIUR Project CLARA (*CLoud Platform for lAndslide Risk Assessment*).

Bibliography

1. H. Adeli and H. Park, A neural dynamics model for structural optimization - theory, *Computers and Structures* **57**(3) (1995) 383–390.
2. H. Adeli and H. Park, Optimization of space structures by neural dynamics, *Neural Networks* **8**(5) (1995) 769–781.
3. F. Ahmadvanlou and H. Adeli, Optimum cost design of reinforced concrete slabs using neural dynamics model, *Engineering Applications of Artificial Intelligence* **18**(1) (2005) 65–72.
4. M. Anderson, Neural reuse: a fundamental organizational principle of the brain, *Beh. Brain Sci.* **33**(4) (2010) 245–266.
5. P. Arena, S. Caccamo, L. Patané and R. Strauss, A computational model for motor learning in insects *IJCNN*, (Dallas, TX, 2013), pp. 1349–1356.
6. P. Arena, M. Calí, L. Patané, A. Portera and R. Strauss, Modeling the insect mushroom bodies: application to sequence learning, *Neural Networks* **67** (2015) 37–53.
7. P. Arena, M. Calí, L. Patané, A. Portera and R. Strauss, A mushroom bodies inspired spiking network for classification and sequence learning, *IJCNN*, (Ireland, 2015), pp. 1–8.
8. P. Arena, S. De Fiore, L. Fortuna, M. Frasca, L. Patané and G. Vagliasindi, Weak chaos control for action-oriented perception: Real time implementation via FPGA, *Biorob*, Feb 2006, pp. 555–560.
9. P. Arena, S. De Fiore, L. Patané, M. Pollino and C. Ventura, STDP-based behavior learning on tribot robot *Proceedings of SPIE* **7365**2009.
10. P. Arena, L. Fortuna, M. Frasca, G. Ganci and L. Patané, A bio-inspired auditory perception model for amplitude-frequency clustering *Proceeding of SPIE* **5839**-2005, pp. 359–368.
11. P. Arena, L. Fortuna, M. Frasca and L. Patané, Sensory feedback in CNN-based central pattern generators, *International Journal of Neural Systems* **13**(6) (2003) 349–362.
12. P. Arena, L. Fortuna, M. Frasca and L. Patané, A cnn-based chip for robot locomotion control, *IEEE Transactions on Circuits and Systems I* **52**(9) (2005) 1862–1871.
13. P. Arena, L. Fortuna, M. Frasca and L. Patané, Learning anticipation via spiking networks: application to navigation control, *IEEE Trans. on Neural Networks* **20**(2) (2009) 202–216.
14. P. Arena, L. Fortuna, M. Frasca, L. Patané and M. Pavone, Climbing obstacles via bio-inspired cnn-cpg and adaptive attitude control *Proceedings - IEEE International Symposium on Circuits and Systems* 2005, pp. 5214–5217.
15. P. Arena, S. Maceo, L. Patané and R. Strauss, A spiking network for spatial memory formation: towards a fly-inspired ellipsoid body model *IJCNN*, (Dallas, TX, 2013), pp. 1245–1250.
16. P. Arena and L. Patané, *Spatial Temporal Patterns for Action Oriented Perception in Roving Robots* (Springer, 2008).
17. P. Arena and L. Patané, Simple sensors provide inputs for cognitive robots, *Instrumentation & Measurement Maga-*

- zine **12**(3) (2009) 13–20.
18. P. Arena and L. Patané, *Spatial Temporal Patterns for Action-Oriented Perception in Roving Robots II: an insect brain computational model* (Springer, Series: Cognitive Systems Monographs, Vol. 21, 2014).
 19. P. Arena and L. Patané, Robotic experiments for sequence learning (2016), online available at <http://www.diees.unict.it/users/parena/biorobots.html>.
 20. P. Arena, L. Patané, V. Stornanti, P. Termini, B. Zaepf and R. Strauss, Modelling the insect mushroom bodies: Application to a delayed match-to-sample task, *Neural Networks* **41** (2013) 202–211.
 21. P. Arena, L. Patané and R. Strauss, The insect mushroom bodies: a paradigm of neural reuse *ECAL*, (MIT Press, Taormina, Italy, 2013), pp. 765–772.
 22. P. Arena, L. Patané and P. Termini, Learning expectation in insects: a recurrent spiking neural model for spatio-temporal representation, *Neural Networks* **32** (2012) 35–45.
 23. P. Arena, L. Patané and P. Termini, Modeling attentional loop in the insect mushroom bodies, *IJCNN*, (Brisbane, Australia, 2012), pp. 7–12.
 24. Y. Aso and et al, Mushroom body output neurons encode valence and guide memory-based action selection in drosophila, *eLife* **3** (2014).
 25. Y. Aso and et al, The neuronal architecture of the mushroom body provides a logic for associative learning, *eLife* **3** (2014).
 26. B. Baddeley, P. Graham, P. Husbands and A. Philippides, A model of ant route navigation driven by scene familiarity, *PLoS Comput Biol* **8**(8) (2012) p. e1002336.
 27. J. Brea, W. Senn and J. Pfister, Matching recall and storage in sequence learning with spiking neural networks, *The Journal of Neuroscience* **33**(23) (2013) 9565–9575.
 28. R. Brooks, A robust layered control system for a mobile robot, *Robotics and Automation, IEEE Journal of* **2**(1) (1986) 14–23.
 29. S. Cassenaer and G. Laurent, Hebbian stdp in mushroom bodies facilitates the synchronous flow of olfactory information in locusts, *Nature* **448**(7154) (2007) 709–13.
 30. L. Chittka and J. Niven, Are bigger brains better?, *Current Biology* **19** (2009) 995–1008.
 31. T. Collet, S. Fry and R. Wehner, Sequence learning by honeybees, *Journal of Comparative Physiology A* **172**(6) (1993) 693–706.
 32. F. Cona and M. Ursino, A multi-layer neural-mass model for learning sequences using theta/gamma oscillations, *International Journal of Neural Systems* **23**(03) (2013) p. 1250036.
 33. R. Davis and K. Han, Neuroanatomy: Mushrooming mushroom bodies, *Current Biology* **6** (1996) 146–148.
 34. P. J. Drew and L. F. Abbott, Extending the effects of spike-timing-dependent plasticity to behavioral timescales, *Proceedings of the National Academy of Sciences* **103**(23) (2006) 8876–8881.
 35. J. B. Duffy, Gal4 system in *Drosophila*: A fly geneticist's swiss army knife, *genesis* **34**(1-2) (2002) 1–15.
 36. B. Duistermars, and M. Frye, Multisensory integration for odor tracking by flying drosophila behavior, circuits and speculation, *Communicative & Integrative Biology* **3**(1) (2010) 60–63.
 37. S. Farris, Are mushroom bodies cerebellum-like structures?, *Arthropod Struct Dev.* **40**(4) (2011) 368–379.
 38. J. Foucaud, J. Burns and F. Mery, Use of spatial information and search strategies in a water maze analog in *Drosophila melanogaster*, *PLoS ONE* **5**(12) (2010).
 39. J. Friedrich, R. Urbanczik and W. Senn, Code-specific learning rules improve action selection by populations of spiking neurons, *International Journal of Neural Systems* **24**(05) (2014) p. 1450002.
 40. S. Ghosh-Dastidar and H. Adeli, Improved spiking neural networks for eeg classification and epilepsy and seizure detection, *International Journal of Neural Systems* **14**(03) (2007) 187–212.
 41. S. Ghosh-Dastidar and H. Adeli, A new supervised learning algorithm for multiple spiking neural networks with application in epilepsy and seizure detection, *Neural Networks* **22**(10) (2009) 1419–1431.
 42. S. Ghosh-Dastidar and H. Adeli, Spiking neural networks, *International Journal of Neural Systems* **19**(4) (2009) 295–308.
 43. M. Giurfa, Cognitive neuroethology: dissecting non-elemental learning in a honeybee brain, *Current opinion in neurobiology* **13**(6) (2003) 726–735.
 44. C. Glenn, B. Maxim and L. Gilles, Olfactory representations by *Drosophila* mushroom body neurons, *Journal of neurophysiology* (2007) p. 734.
 45. W. Gronenberg and G. Lopez-Riquelme, Multisensory convergence in the mushroom bodies of ants and bees, *Acta. Biol. Hung.* **55** (2004) 31–37.
 46. N. Gupta and M. Stopfer, Functional analysis of a higher olfactory center, the lateral horn, *The Journal of Neuroscience* (2012) p. 8138.
 47. R. Huerta, T. Nowotny, M. Garcia-Sanchez, H. Abarbanel and M. Rabinovich, Learning classification in the olfactory system of insects, *Neural Comput.* **16**(8) (2004) 1601–40.
 48. R. Huerta, S. Vembu, J. M. Amigó, T. Nowotny and C. Elkan, Inhibition in multiclass classification, *Neural Comput.* **24** (September 2012) 2473–2507.
 49. E. M. Izhikevich, Which model to use for cortical spiking neurons?, *IEEE Transactions on Neural Networks* **15**(5) (2004) 1063–1070.
 50. E. M. Izhikevich, Solving the distal reward problem through linkage of stdp and dopamine signaling, *Cerebral Cortex* **17** (2007) 2443–2452.
 51. H. Jaeger, Short term memory in echo state networks, *GMD-Report German National Research Institute for Computer Science* **152** (2002).
 52. O. Jensen and J. E. Lisman, Hippocampal sequence-encoding driven by a cortical multi-item working memory buffer, *Trends in Neurosciences* **28**(2) (2005) 67 – 72.
 53. W. Lewinger and R. Quinn, Neurobiologically-based control system for an adaptively walking hexapod, *Industrial Robot* **38**(3) (2011) 258–263.
 54. X. Liu and R. Davis, Insect olfactory memory in time and space, *Curr. Opin. Neurobiol.* **6** (2006) 679–685.

55. N. R. Luque, J. A. Garrido, R. R. Carrillo, S. Tolu and E. Ros, Adaptive cerebellar spiking model embedded in the control loop: Context switching and robustness against noise, *International Journal of Neural Systems* **21**(05) (2011) 385–401.
56. W. Maass, T. Natschlaeger and H. Markram, Real-time computing without stable states: A new framework for neural computation based on perturbations, *Neural Computation* **14**(11) (2002) 2531–2560.
57. R. May and A. Wellman, Alternation in the fruit fly *Drosophila melanogaster*, *Neuroscience and biobehavioral reviews* **12**(Psychonomic science) (1968) 339–340.
58. J. L. McKinstry and G. M. Edelman, Temporal sequence learning in winner-take-all networks of spiking neurons demonstrated in a brain-based device, *Frontiers in Neurobotics* **7**(10) (2013).
59. A. Mohemmed, S. Schliebs, S. Matsuda and N. Kasabov, Span: Spike pattern association neuron for learning spatio-temporal spike patterns, *International Journal of Neural Systems* **22**(04) (2012) p. 1250012.
60. C. Morris and H. Lecar, Voltage oscillations in the barnacle giant muscle fiber, *Biophys. Journal* **35** (1981) 193–213.
61. T. S. Mosqueiro and R. Huerta, Computational models to understand decision making and pattern recognition in the insect brain, *Curr. Op. in Insect Science* **6** (2014) 80 – 85.
62. R. Murphey, Sequential alternation behavior in the fruit fly *Drosophila melanogaster*, *Journal of Comparative Physiology and Psychology* **60**(2) (1965) 196–199.
63. K. Neuser, T. Triphan, M. Mronz, B. Poeck and R. Strauss, Analysis of a spatial orientation memory in drosophila, *Nature* **453** (2008) 1244–1247.
64. E. Nichols, L. J. McDaid and N. H. Siddique, Case study on a self-organizing spiking neural network for robot navigation, *International Journal of Neural Systems* **20**(06) (2010) 501–508.
65. J. Niven and L. Chittka, Reuse of identified neurons in multiple neural circuits, *Behavioral and Brain Sciences* **33**(4) (2010) p. 285.
66. T. Nowotny, R. Huerta, H. Abarbanel and M. Rabinovich, Self-organization in the olfactory system: one shot odor recognition in insects., *J. Comput. Neurosci.* **93** (2005) 436–446.
67. T. Nowotny, M. Rabinovich, R. Huerta and H. Abarbanel, Decoding temporal information through slow lateral excitation in the olfactory system of insects, *J. Comput. Neurosci.* **15** (2003) 271–281.
68. T. Nowotny, M. I. Rabinovich and H. D. I. Abarbanel, Spatial representation of temporal information through spike-timing-dependent plasticity, *Phys. Rev. E* **68** (Jul 2003) p. 011908.
69. H. Park and H. Adeli, A neural dynamics model for structural optimization - application to plastic design of structures, *Computers and Structures* **57**(3) (1995) 391–399.
70. L. Patané, S. Hellbach, A. F. Krause, P. Arena and V. Duerr, An insect-inspired bionic sensor for tactile localisation and material classification with state-dependent modulation, *Frontiers in Neurobotics* **6**(8) (2012).
71. J. Perez-Orive, O. Mazor, G. Turner, S. Cassenaer, R. Wilson and G. Laurent, Oscillations and sparsening of odor representations in the mushroom body, *Science* **297** (2002) 359–365.
72. S. Pick and R. Strauss, Goal-driven behavioral adaptations in gap-climbing *Drosophila*, *Curr. Biol.* **15** (2005) 1473–8.
73. J. Rossello, V. Canals, A. Oliver and A. Morro, Studying the role of synchronized and chaotic spiking neural ensembles in neural information processing, *International Journal of Neural Systems* **24**(5) (2014) 1–11.
74. P. Russo, B. Webb, R. Reeve, P. Arena and L. Patané, Cricket-inspired neural network for feedforward compensation and multisensory integration, *IEEE Conference on Decision and Control*, 2005.
75. S. Scherer, R. Stocker and B. Gerber, Olfactory learning in individually assayed *Drosophila* larvae, *Learn. Mem.* **10** (2003) 217–225.
76. M. Schmuker, T. Pfeil and M. P. Nawrot, A neuromorphic network for generic multivariate data classification, *Proceedings of the National Academy of Sciences* **111**(6) (2014) 2081–2086.
77. A. Schneider, J. Paskarbeit, M. Schilling and J. Schmitz, Hector, a bio-inspired and compliant hexapod robot, *Biomimetic and Biohybrid Systems*, eds. A. Duff, N. Lepora, A. Mura, T. Prescott and P. Verschure, *Lecture Notes in Computer Science* **8608** (Springer International Publishing, 2014), pp. 427–429.
78. S. Shapero, M. Zhu, P. Hasler and C. Rozell, Optimal sparse approximation with integrate and fire neurons, *International Journal of Neural Systems* **24**(5) (2014) 1–16.
79. S. Song, K. D. Miller and L. F. Abbott, Competitive hebbian learning through spike-timing-dependent plasticity, *Nature Neurosci.* **3** (2000) 919–926.
80. R. Stocker, C. Lienhard and A. Borst, Neuronal architecture of the antennal lobe in *drosophila melanogaster*, *Cell Tissue Res* (1990) 9–34.
81. M. Stopfer, Central processing in the mushroom bodies, *Current Opinion in Insect Science* **6**(0) (2014) 99 – 103.
82. B. Strack, K. Jacobs and K. Cios, Simulating vertical and horizontal inhibition with short term dynamics in a multi-column, multi-layer model of neocortex, *International Journal of Neural Systems* **24**(5) (2014) 1–19.
83. T. J. Strain, L. J. McDavid, T. M. McGinnity, L. P. Maguire and H. M. Sayers, An STDP training algorithm for a spiking neural network with dynamic threshold neurons, *Int. Journal of Neural Systems* **20**(06) (2010) 463–480.
84. N. J. Strausfeld, Organization of the honey bee mushroom body: Representation of the calyx within the vertical and gamma lobes, *The Journal of Comparative Neurology* **450**(1) (2002) 4–33.
85. R. Strauss, The central complex and the genetic dissection of locomotor behaviour, *Curr. Opin. Neurobiol.* **12** (2002) 633–638.
86. H. Tanimoto, M. Heisenberg and B. Gerber, Event timing turns punishment to reward, *Arthropod Struct. Dev.* **430** (2004) p. 983.
87. B. Webb, J. Wessnitzer, S. Bush, J. Schul, J. Buchli and A. Ijspeert, Resonant neurons and bushcricket behaviour, *Journal of Comp. Phys. A* **193**(2) (2007) 285–288.

88. M. Wehr and G. Laurent, Odor encoding by temporal sequences of firing in oscillating neural assemblies, *Nature* **384** (1996) 162–166.
89. J. Wessnitzer, J. Young, J. Armstrong and B. Webb, A model of non-elemental olfactory learning in *Drosophila*, *J. Neurophysiol.* **32** (2012) 197–212.
90. G. Zhang, H. Rong, F. Neri and M. Perez-Jimenez, An optimization spiking neural p system for approximately solving combinatorial optimization problems, *International Journal of Neural Systems* **24**(5) (2014) 1–16.
91. S. Zhang, A. Si and M. Pahl, Visually guided decision making in foraging honeybees, *Frontiers in Neuroscience* **6**(88) (2012) 1–17.
92. S. Zhang, K. Bartsch and M. Srinivasan, Maze learning by honeybees, *Neurobiology of learning and memory* **66**(3) (1996) 267–282.