# Insect inspired spatial-temporal cellular processing for feature-action learning

Paolo Arena, Luca Patané, Angelo Spinosa
DIEEI - University of Catania
viale A. Doria 6 Catania, Italy
Email: [parena, lpatane]@dieei.unict.it

*Abstract*—**In this paper an insect brain-inspired neural processing architecture was developed to be applied on board of a bio-robot for solving feature-to-action tasks. The system, accounting on visual features, is able to solve a classification problems using a spatial temporal approach that is typical of bio-inspired neural architectures. The proposed neural structure, taking inspiration from a specific neuropile of the insect brain, called mushroom bodies, is applied to solve tasks shown in insect experiments where non-elemental learning strategies are taken into account. An important peculiarity of the hidden processing layer of the proposed multi-layer architecture is the local, CNN-like connectivity among the spiking neurons, opening the way for an hardware implementation on neuromorphic chips.**

## I. Introduction

Classification is one of the most important tasks performed by living beings while processing information coming from the environment. Current researches in Robotics are focused on the development of autonomous systems able to solve tasks such as object recognition and decision making in structured and unstructured environments. In previous works we developed an insect-inspired architecture to model several different behaviours shown by insects and, in particular, by *Drosophila melanogaster* that is a perfect model organism [1], [2], [3]. One of the first model for olfactory learning, inspired by the Mushroom Bodies (MBs), was introduced in [4]. This model focused on an associative leaning-based structure where the input-driven MBs activity is associated to a rewarding or a punishing event. In [5] spatio-temporal inputs were encoded into the neural structure that produced spatial patterns. This sparse activity can be associated to different classes using a reinforcement learning approach, obtaining a structure similar to a support vector machine [6]. More recent works focalized on the clustering capabilities of an MB-inspired network, proposing a spiking neural network for the classification of multivariate data: however, here an additional processing stage using a neural gas architecture was requested[7]. With respect to this structure, in this work we propose a much simpler multi-layers network of spiking neurons that takes inspiration by the processing flow within the MBs in the fruit fly and avoids using Neural gas classifiers which make the classification task much easier in [7]. The architecture here used represents a part of a more complex structure that has been previously introduced in [3], [8]. In this work the classification capabilities of the proposed neural architecture in solving problems of non-elemental learning, that were demonstrated to be affordable also by insects [9], are evaluated. When modelling complex neural activities to generate time-varying signals, two possible strategies can be followed: the use of enslaved chaotic dynamics [10] and the exploitation of Reservoir Computation by extracting the needed dynamics using read-out maps [11], [12], [13]. Focusing our attention on the second approach, our computational model is here used as a neural controller for a fly-inspired simulated walking robot facing with different scenarios inspired by experiments performed with honeybees [14], [15]. These insects are able to extract visual features from objects to perform decision making processes while negotiating a complex environment. Several different experiments were performed to evaluate the capabilities of insects to respond to stimuli through a decision making process. For instance Giurfa and co-workers trained honeybees to fly into a Y-maze to evaluate when different learning strategies, as elemental or configural visual discrimination, occur on the basis of the number of trials [16]. Honeybees can learn visual stimuli during food search and can solve visual discrimination problems that contain ambiguity at the feature level. In particular, in [15], three different visual discriminations were considered: positive patterning; negative patterning and biconditional discrimination. In all the three cases a non-elemental processing of stimulus compounds is needed in order to learn how to solve the problem. Similar experiments were reproduced in this paper to evaluate the performance of the developed network that has been implemented using GeNN (GPU-enhanced Neuronal Network simulation environment). It is a code generation environment for developing high performing simulations of brain-inspired neural circuits exploiting the parallel computational processing of GPU [17]. The neural controller was interfaced with a dynamic simulation environment [18] where a fly-inspired walking robot was developed and tested.

## II. Neural Model

The proposed neural architecture consists of four distinct layers as depicted in Fig. 1. The first layer is an interface needed between the acquired input patterns and the neural network. It introduces in the input layer a reasonable level of current to efficiently encode the features of the object/event to be classified. The input layer contains a number of class I Izhikevich spiking neurons [19] equal to the input features to be processed. The core of the architecture is constituted

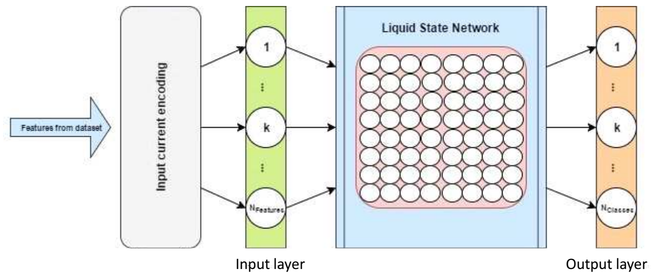


Fig. 1. Block scheme of the MB-inspired developed architecture devoted to classification. The external input is pre-processed by a layer that randomly excite the lattice of locally connected spiking neurons. The spiking activity is exploited for classification using multiple read-out maps in the output layer that are trained to distinguish the different presented classes.

by a lattice of locally connected spiking neurons of the same type as the input layer, making this structure in between a CNN and a liquid state network [12]. At this level we have a boost of dimensions, needed to correctly classify non-linear separable patterns. In the last layer we have a series of neurons working as spatial integrators, to collect the activity distributed on the lattice, providing in output a weighted sum of the neural activity. The number of output neurons is equal to the number of classes/decisions to be identified/selected. Details on the model are reported in the following.

### A. Input layer and current encoding

Each input pattern is a vector of features to be processed and associated to the available classes: $\mathbf{P}_j = [p_{1,j}, p_{2,j} ..., p_{n,j}]$.

The input encoding procedure is reported below:

1) the range of variability, of input currents $[I_{min}, I_{max}]$ is chosen so as to allow input neurons fire and create a significant neural activity within the liquid layer;
2) to get a normalized input current, the maximum $(M_k)$ and the minimum $(m_k)$ values of each feature $k$ are calculated;
3) if the current applied to the $k$-th input neuron from the $j$-th input pattern is $I_{k,j}$, then the normalized input value is evaluated as:

$$I_{k,j} = I_{min} + I_{max}\frac{p_{k,j} - m_k}{M_k - m_k} \tag{1}$$

### B. The CNN-Liquid State Layer

The weights coming from the input layer to the lattice of spiking neurons (LSN) are fixed to 1 and subject to a $25\%$ probability of being connected to the LSN. The lattice contains a percentage of $75\%$ excitatory and $25\%$ inhibitory neurons [8]. The synaptic weight values among the neurons are fixed and can randomly assume $-10$ or 10. The generation of the synaptic connections within the lattice is based on a probability depending on the distance $d_{i,j}$ between the presynaptic $(i)$ and postsynaptic $(j)$ neurons (see [1] for more details). The presence of synaptic connections within the liquid lattice depends on the euclidean distance $d_{i,j}$ between the pre-synaptic $(i)$ and post-synaptic $(j)$ neurons within the LSN. The synaptic model is a typical impulse response used to generate an input current for the post-synaptic neuron when the pre-synaptic neuron emits a spike. The time constant of the synaptic model was randomly chosen among the values $\tau$ = 5, and $\tau$ = 15 $ms$. This variability has shown to improve the dynamics generated inside the network within the processing time window. The number of neurons considered in this layer is below 100 (e.g. $8x8$, $9x9$) that will be demonstrated to be sufficient to obtain positive results. Therefore, the network contains a minimal number of neurons able to show the learning skills and behavioural responses as in the biological case of the insect model organism. Basically we are trying to identify the core of the structure useful for classification that can be enlarged in terms of number of neurons and connections if we need to improve the network capabilities. The output neurons, modelled with a linear transfer function, are massively connected with the LSN. Finally the output weights, representing the read-out map are subject to learning.

### C. Target signals for classification

Classification is here treated as a supervised learning procedure. Once defined the input patterns, composing a learning set $S$, we need to define the target signals for each input to be associated to a specific class. Using the same strategy employed for motor-skill learning [20] we chose, as target signals, simple exponential functions of the type $T(t) = 1 - e^{\frac{-t}{\tau_t}}$, with two different time constants $\tau_t$, one much higher than the other, at the aim to represent two dynamics with different speeds.

In particular, if an enhanced response is desired (i.e. for the class corresponding to the currently considered input) the output signal has to reach the maximum value equal to one in the time window of $50ms$ (corresponding to 500 integration steps with an integration step $dt = 0.1ms$): this is simply obtained using a small time constant $\tau_t = 1ms$. On the other hand, the target signal is confined to low values using a larger time constant $\tau_t = 100ms$.

### D. Learning strategy

The classification task is accomplished by implementing a learning algorithm. In particular, the free parameters to be learned are collected in the synaptic weight matrix $\mathbf{W}$, which is related to the vector $\mathbf{T}$, representing the target signal, by the following equation: $\mathbf{T} = \mathbf{ZW}$.

Generally, $\mathbf{Z}$ is a rectangular matrix comprising all the time-varying post-synaptic currents coming from the lattice and directed to the output neurons. We initially investigated the performance of the architecture adopting a standard method to solve the problem: the Moore-Penrose pseudoinverse of $\mathbf{Z}$ is calculated $(\mathbf{Z}^+)$, thus obtaining $\mathbf{Z}^+ \equiv \left(\mathbf{Z}^T\mathbf{Z}\right)^{-1}\mathbf{Z}^T$, and then $\mathbf{W} = \mathbf{Z}^+\mathbf{T}$.

Simple incremental learning strategies, based on the Least Mean Square algorithm, adopted in other works [8], are currently being evaluated for further development.

## III. Case of Study

Classification problems can be related not only to object recognition but also to decision making processes. For instance

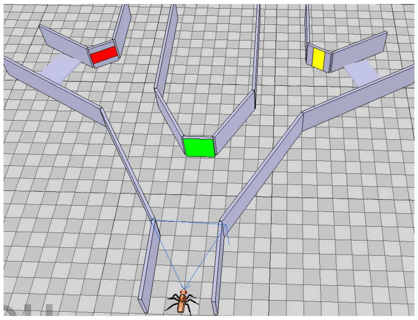

Fig. 2. Simulation set-up developed in the V-REP dynamic environment. A fly-inspired legged robot is asked to walk in a multiple Y-maze environment selecting the suitable route and motor actions through the processing of visual cues acquired from landmarks placed in each selection chamber.

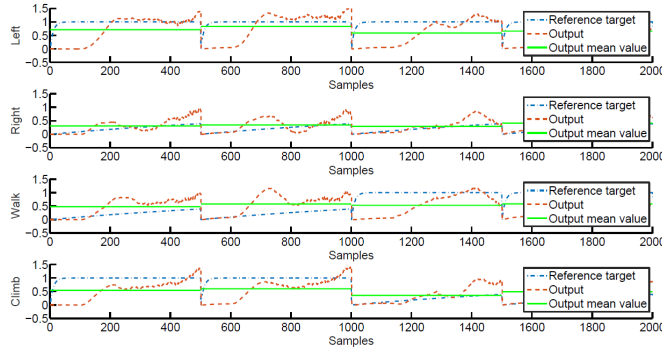


Fig. 3. Comparison between the assigned target, the output of the network after the learning process and its mean value.

motor actions can be performed according to the environmental conditions as demonstrated in maze experiments with honeybees [15]. To reproduce these experiments, a multiple Y-maze was designed. The fly-inspired walking robot should solve the maze by processing some visual landmarks, whose properties provide the required information to make two distinct and independent decisions: select the turning direction and decide a behaviour (i.e. walking or climbing), based on the presence of an obstacle in the chosen direction. The robotic system and the working scenario were implemented in the dynamic simulation environment [18] as depicted in Fig. 2

Locomotion control is performed using a Central Pattern Generator developed through a multi-template approach [21]. The robot is equipped with proximity sensors used to avoid collisions [22], [23] and with a visual system able to extract visual cues from landmarks placed in each chamber of the maze. Each landmark is endowed with two types of information: colour and shape. The first property is encoded as an RGB vector representing the chromatic intensities for each channel, whereas the shape can be set in few configurations (square, vertical or horizontal rectangle). The former feature is used to encode the turning direction, the latter to codify the presence of an obstacle in the right or left branch. Furthermore, following the biological experiments [15], we performed a simulation related to Positive Patterning Discrimination: given two features A and B, the paradigm assumes that either A or B are reinforcing features, while the combined feature AB is not reinforced. In our simulations, the network has been designed to provide in output two independent set of motor actions: ”Turn-left” or ”Turn-Right” and ”Walk” or ”Climb”. The left and right decision depends on the colour of the landmark: in the performed simulation both the red and green colour were associated to a right turn whereas the combination of the two (i.e. yellow) is associated to a left turn creating a non linear separable problem as depicted in [15]. The shape of the landmark is also used: the horizontal/vertical distributedness is a common feature used by insects to discriminate objects. In our case a prominence in the horizontal (vertical) distributedness means an obstacle on the right (left) side of the Y-maze. The square shape indicates the absence of obstacles. If an object is present in the selected branch, the locomotion is adapted to allow the climbing [24]. The learning dataset is constituted by nine elements, five for the input and four for the target. The five inputs correspond to the RGB component of the landmark and the horizontal and vertical distributedness that were simplified in this simulation as digital values on the basis of a threshold on the base-to-height ratio of the acquired rectangular shape of the landmark. The blue channel does not contain useful information for the designed experiment: therefore it introduce noise into the network. Several network configurations were randomly generated and tested using a dataset of 2750 entries: the $80\%$ were used for the learning phase and the remaining for the testing phase. An example of comparison between the assigned target and the output generated by the network is reported in Fig. 3. The network outputs are evaluated couple-wise (Turn-Left vs Turn-right and Walk vs climb). To select the two winning output neurons for each input pattern, a simple analysis is performed evaluating the mean value of the time evolution during the 500 integration steps of simulation for each couple of neurons. The output neurons with the highest level, evaluated couple-wise, win the competition and will drive the robot actions.

Several performance indexes can be calculated for evaluation. An approach, commonly used, is based on the confusion matrices. For a multi-class classification issue (like in this case) with $N$ output classes, a confusion matrix is a $N$-by-$N$ matrix that quantifies how many classes have been correctly identified. The diagonal of this matrix reports the positive results, whereas all the other terms are classification errors; therefore, if the confusion matrix is $\mathbf{M} = (M_{ij})$ then the success rate is given by:

$$P_{\text{success}} = \frac{\text{Tr}\,(\mathbf{M})}{\sum_{i,j} M_{ij}} \times 100 \qquad (2)$$

Another common index considered in machine learning is the Matthews Correlation Coefficient [25] defined as follows:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)\,(TP + FN)\,(TN + FP)\,(TN + FN)}}$$

where $TP, TN, FP, FN$ are the number of true positives, true negatives, false positives and false negatives, respectively.

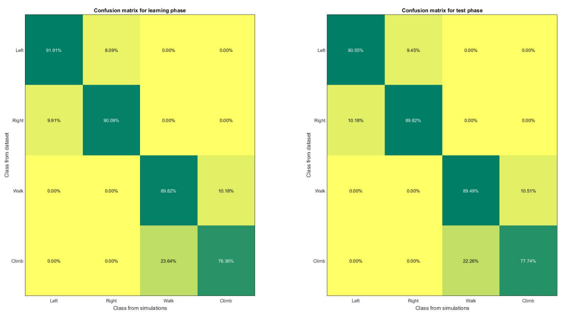


Fig. 4. Confusion matrix obtained from the simulation of a representative case both for the learning patterns and for the testing ones.

The results obtained in the performed experiments are the following:

$$P_{\text{success},learning} = 87.05\%; MCC_{\text{learning}} = 0.82;$$
$$P_{\text{success},test} = 86.91\%; MCC_{\text{test}} = 0.82$$

The confusion matrix obtained for the learning and the testing phase are also reported in Fig. 4. The results obtained allow to efficiently drive the robot model and are prone to be further improved, especially in the fourth choice: the testing results are at a lower level than the learning ones. This could be due to the absence of enough information into the learning set, and this is currently under investigation.

## IV. Conclusion

In this work we propose a neural structure inspired by the insect MBs, able to perform pattern discrimination also in presence of non-elementary association. From biological experiments with honeybees a similar simulation was set up, considering a *Drosophila*-inspired walking robot moving in a Y-maze environment. Standard performance indexes were considered to evaluate the proposed neural architecture. Notably, the structure used here for classification can be integrated within a more complex neuro-computing architecture, inspired by the insect brain and reported in literature, able to show other complex behaviors, like attention, expectation, motor and sequence learning [26], [8]. For instance, the output neurons can be used for different purposes: to classify inputs as well as to provide time varying signals to control relevant parameters of the locomotion system in case of motor learning [27].

## Acknowledgment

This work was partially supported by the UNICT-FIR 2014 Project and by MIUR Project CLARA.

## References

[1] P. Arena, M. Calí, L. Patané, A. Portera, and R. Strauss, "Modelling the insect mushroom bodies: Application to sequence learning," *Neural Networks*, vol. 67, pp. 37–53, 2015.

[2] P. Arena, C. Berg, L. Patané, R. Strauss, and P. Termini, "An insect brain computational model inspired by drosophila melanogaster: Architecture description," in *Proceedings of the International Joint Conference on Neural Networks*, Barcelona, Spain, 2010, pp. 831–837.

[3] P. Arena, S. Caccamo, L. Patané, and R. Strauss, "A computational model for motor learning in insects," in *Proceedings of International Joint Conference on Neural Networks*, Dallas, TX, 2013, pp. 1349–1356.

[4] B. Gerber, H. Tanimoto, and M. Heisenberg, "An engram found? evaluating the evidence from fruit flies," *Current Opinion in Neurobiology*, vol. 14, pp. 737–744, 2004.

[5] 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.*, vol. 15, pp. 271–281, 2003.

[6] R. Huerta, S. Vembu, J. M. Amigó, T. Nowotny, and C. Elkan, "Inhibition in multiclass classification," *Neural Comput.*, vol. 24, no. 9, pp. 2473–2507, Sep. 2012.

[7] M. Schmuker, T. Pfeil, and M. P. Nawrot, "A neuromorphic network for generic multivariate data classification," *PNAS*, vol. 111, no. 6, pp. 2081–2086, 2014.

[8] P. Arena, M. Calí, L. Patané, A. Portera, and R. Strauss, "A fly-inspired mushroom bodies model for sensory-motor control through sequence and subsequence learning," *Int. J. Neural Syst.*, vol. 26, no. 6, pp. 1–18, 2016.

[9] J. Wessnitzer, J. Young, J. Armstrong, and B. Webb, "A model of non-elemental olfactory learning in *Drosophila*," *J. Neurophysiol.*, vol. 32, pp. 197–212, 2012.

[10] P. Arena, S. De Fiore, L. Fortuna, and L. Patané, "Perception-action map learning in controlled multiscroll systems applied to robot navigation," *Chaos*, vol. 18, no. 4, 2008.

[11] H. Jaeger, "Short term memory in echo state networks," *GMD-Report German Nat. Res. Inst. for Computer Science*, vol. 152, 2002.

[12] W. Maass, T. Natschlaeger, and H. Markram, "Real-time computing without stable states: A new framework for neural computation based on perturbations." *Neural Computation*, vol. 14(11), pp. 2531–2560, 2002.

[13] P. Arena, L. Fortuna, D. Lombardo, and L. Patané, "Perception for action: Dynamic spatiotemporal patterns applied on a roving robot," *Adaptive Behavior*, vol. 16, no. 2-3, pp. 104–121, 2008.

[14] S. Zhang, A. Si, and M. Pahl, "Visually guided decision making in foraging honeybees," *Front. Neurosci.*, vol. 6, no. 88, pp. 1–17, 2012.

[15] M. Schubert, H. Lachnit, S. Francucci, and M. Giurfa, "Nonelemental visual learning in honeybees," *Animal Beh.*, vol. 64, pp. 175–184, 2002.

[16] M. Giurfa, M. Schubert, C. Reisenman, B. Gerber, and H. Lachnit, "The effect of cumulative experience on the use of elemental and configural visual discrimination strategies in honeybees," *Beh. Brain Res.*, vol. 145, no. 1-2, pp. 161–169, 2003.

[17] E. Yavuz, J. Turner, and T. Nowotny, "Genn: a code generation framework for accelerated brain simulations," *Scientific Reports*, vol. 6, no. 18854, 2016.

[18] E. Rohmer, S. P. N. Singh, and M. Freese, "V-rep: A versatile and scalable robot simulation framework," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 1321–1326.

[19] E. Izhikevich, "IEEE which model to use for cortical spiking neurons?" *Transactions on Neural Networks*, vol. 15(5), pp. 1063–1070, 2004.

[20] E. Arena, P. Arena, R. Strauss, and L. Patané, "Motor-skill learning in an insect inspired neuro-computational control system," *Frontiers in Neurorobotics*, vol. 11, p. 12, 2017.

[21] P. Arena, L. Fortuna, and M. Frasca, "Multi-Template Approach to realize central pattern generators for artificial locomotion control," *Int. J. Circ. Theor. Appl.*, vol. 30, pp. 441–458, 2002.

[22] P. Arena, L. Fortuna, M. Frasca, and L. Patané, "Sensory feedback in CNN-based central pattern generators," *International Journal of Neural Systems*, vol. 13(6), pp. 349–362, December 2003.

[23] P. Arena and L. Patané, "Simple sensors provide inputs for cognitive robots," *Instr. & Meas. Magazine*, vol. 12(3), pp. 13–20, 2009.

[24] E. Arena, P. Arena, R. Strauss, and L. Patan, "Motor-skill learning in an insect inspired neuro-computational control system," *Frontiers in Neurorobotics*, vol. 11, p. 12, 2017.

[25] B. W. Matthews, "Comparison of the predicted and observed secondary structure of t4 phage lysozyme," *Biochimica et Biophysica Acta*, vol. 405, no. 2, pp. 442–451, 1975.

[26] P. Arena, L. Patané, and P. Termini, "Learning expectation in insects: A recurrent spiking neural model for spatio-temporal representation," *Neural Networks*, vol. 32, pp. 35–45, 2012.

[27] P. Arena, S. Caccamo, L. Patané, and R. Strauss, "A computational model for motor learning in insects," in *International Joint Conference on Neural Networks (IJCNN)*, Dallas, TX, August 4-9, 2013, pp. 1349–1356.