An insect brain computational model inspired by *Drosophila melanogaster*: simulation results

Paolo Arena, Luca Patané and Pietro Savio Termini

***Abstract*— Since many years insects have been considered as a source of inspiration for robotic architectures. From this point of view the fly *Drosophila melanogaster* is more than likely a protagonist, because of the genetic techniques that allow neurobiologists to make deep studies and hypotheses about the brain of this fly. In this work a computational model of the *Drosophila* has been tested and implemented on a robot simulator. Moreover, the normal capabilities of the fly have been extended in order to have an useful robot-oriented model. Results about a possible application in a real-life scenario of the whole model of the *Drosophila* brain are reported.**

I. Introduction

IN the last years animals have considerably influenced robotics. Robotics can improve the research in biology [1] and, moreover, without any doubts results in neurobiology helped the design of robotic structures and control systems [2], [3], [4].
In particular, insects attract the interest because the relative small number of neurons in their brain contrasts with their behavioral complexity. Insect strategies in problem solving have been analyzed and adapted to the implementation in mobile robots [5], [6]. In this way, the modeling of insects brain functionalities seems to be really useful in order to develop efficient robots.

The fly *Drosophila melanogaster* has become the reference point in the insect brain modeling, thanks to the genetic techniques that allow a deep functional analysis of its brain neuropils. Brain structures and functionalities of this fly have been modeled and implemented on real robots [7], [8]. Moreover, the *Drosophila* behavior inspired control systems useful for robotic applications [9].

The goal of this work is to simulate and test a computational architecture presented by the authors in [10]. This model is inspired by recent experiments on the study of the *Drosophila* brain. Moreover, the proposed appoach shows the advantage to be potentially improved and generalized to be useful for real robotic applications.

An overview of the considered computational architecture of the *Drosophila melanogaster* brain is presented in Section II, while Section III and IV describe the simulation setup and the obtained results used to validate the model. A possible real life application of the computational model is finally reported in Section V.

Paolo Arena, Luca Patané and Pietro Savio Termini are with the Dipartimento di Ingegneria Elettrica, Elettronica e dei Sistemi, Universitá degli Studi di Catania, Italy (email: {parena, lpatane, pstermini}@diees.unict.it).

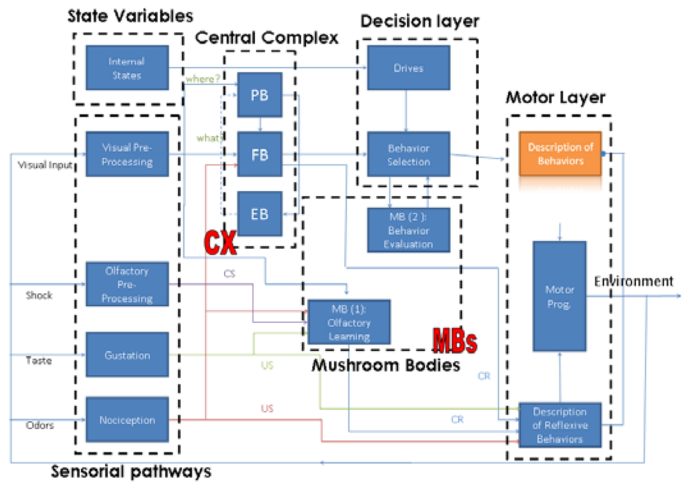

Fig. 1. Computational architecture of the *Drosophila melanogaster* brain. See [10] for details.

II. A sketch about the *Drosophila* brain computational model

The computational model of the *Drosophila* brain, illustrated in [10], is shown in Fig. 1. The model has been designed analyzing biological experiments regarding the olfactory conditioning [12], [13], [14], the visual learning [15] and the orientation memory [16] in real flies. The mainly involved structure of the *Drosophila* brain are Central Complex (CX) and the Mushroom Bodies (MBs). These neuropils have been functionally analyzed in [17], [18], [19]. In this paper a computational model of this relevant structures is proposed. The visual pathway mainly involves the CX model through the ellipsoid body (EB), the fan-shaped body (FB) and the protocerebral bridge (PB) models, while the olfactory pathway goes through the sub model MB(1) of the MBs. The olfactory classical conditioning has been modeled by the Spike Timing Dependent Plasticity (STDP) algorithm [20], [21]. The *Behavior Selection Network* (BSN) is a spiking neural network that allows the robot to select the most suitable behavior in order to satisfy its *Drives*. Drives are related to the internal state of the robot. The *behavior evaluation model* is a mathematical block, inspired from the MBs functionalities [22], that excites (or inhibits) the ongoing behavior if it is being successful (or not). A detailed description of the model can be found in [10].

III. Simulation Setup

The target of this work is to simulate and analyze the capabilities of the proposed computational architecture. Details about the robotic simulator and the implementation of the simulation environment are presented in this Section.

A. The robot and the simulator

The robot used in the experiments is a Pioneer P3-AT differential-drive roving robot. The platform operates as a server in a client-server environment; the onboard PC is used to host the control architecture. The robot dimensions are 50cm x 49cm x 26cm and its weight is 9 Kg.

MobileSim (http://robots.mobilerobots.com) is the software used for simulating the Pioneer P3-AT roving robot in a virtual 2D environment. This simulation environment, that is extremely realistic (e.g. odometry errors are taken into consideration), has been used to evaluate the performance of the proposed control system.

B. Implementation of odors, punishments and rewards in the simulator

In order to implement olfactory classical conditioning it is necessary for the robot to have sensors that can detect odors and that can monitor rewards or punishments given to the robot. In a simulation environment it is convenient to implement virtual sensors. For instance, if an object releases an odor called Odor1, it is convenient to assume the output of the olfactory sensor as a Gaussian function of the distance d from the robot to that object:

$$f_{od}(d) = K_{od} e^{-d/\tau_{od}} \tag{1}$$

where K_{od} is a constant gain and τ_{od} represents the decay of the sensor output when the robot moves away from the object.
It is possible to use a similar strategy to determine the output of a punishment sensor and the output of a reward sensor:

$$f_{pun}(d) = K_{pun} e^{-d/\tau_{pun}} \tag{2}$$

$$f_{rew}(d) = K_{rew} e^{-d/\tau_{rew}} \tag{3}$$

The values of the constants can be determined in order to obtain a tighter Gaussian function for the output of the punishment and reward sensors: in this way, if the robot is approaching the object, it will first detect the odor and then it will be rewarded or punished if that object is not neutral.

IV. Simulation Results

This section presents the experiments made in order to validate each model of the general computational architecture of the *Drosophila melanogaster* brain.

A. Mushroom Bodies and olfactory learning

The following simulation shows how the MBs model for odor learning works. This model is shown in Fig. 2. Each neuron is an Izhikevich Class I spiking neuron [23]. The simulation of the model has been done using the Euler integration method with a constant integration time of 20 milliseconds. The synapses time constant is 800 milliseconds and synaptic weights are initialized to the value of 0.05. A priori known information is codified in the fixed synaptic weights that have been initialized to the value of 10 (excitatory) and -3 (inhibitory). A decay rate has been introduced: every 100 simulation steps all synaptic weights are decreased by 1% percent of their value. The implemented network is the same as shown in Fig. 2. This simulation was performed to verify the capability of the MBs model to make the right associations between odors and rewards or punishments in a complex environment. The robot is introduced into a square arena, 10m x 10m, in which four objects are present. There is an odor spreading out from each object in the environment. In particular, Eq. 1 has been used. Two different odors are associated to these objects and a reward or a punishment is given to the robot when one of the objects is reached, following the association reported in Table I.

Exploring the environment shown in Fig. 3, the robot has to learn that there is a strong association between the Odor 1 and the punishment: in a testing phase, the robot will be able to escape when detecting that odor, before the shock occurs. The behavior of the network neurons during the simulation is shown in Fig. 4 while Fig. 5 presents the trend of the synaptic weights during the simulation. The network evolves for 100 simulation steps for each robot action. At the end of the simulation, the robot has explored the environment completely and it is able to make the right association. Other experiments were performed, obtaining similar results.

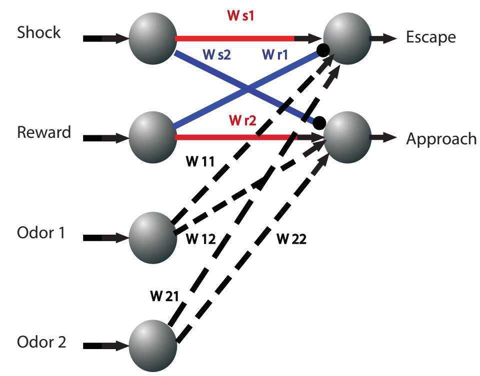

Fig. 2. Olfactory Learning Model. solid line (dashed) connections correspond to fixed (plastic) synapses; arrows (bullet) correspond to excitatory (inhibitory) connections.

Object	Odor	P/N/R
Object A	Odor 1	Punishment
Object B	Odor 2	Reward
Object C	Odor 2	Punishment
Object D	Odor 2	Neutral

TABLE I
Summary of the characteristic of the objects in the MBs model test.

B. Protocerebral bridge and fan-shaped body

Through a functional analysis of the *Drosophila* protocerebral bridge and fan-shaped body, it is possible to suppose

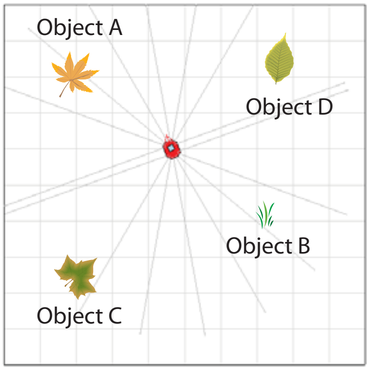

Fig. 3. Simulation environment used for the MB model test.

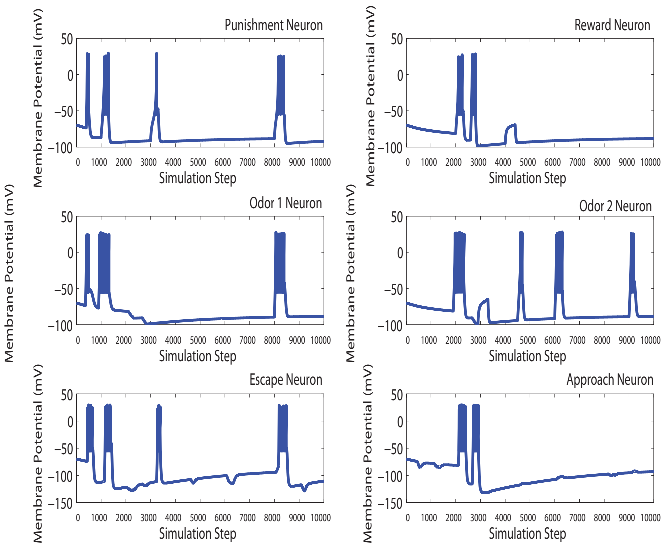

Fig. 4. Simulation results of the olfactory learning model: behavior of the neurons membrane potential during simulation. The implemented network is shown in Fig. 2.

that object detection and distance estimation are mainly performed by the PB, while the FB is related to feature extraction and classification. In the following experiments, the properties of the PB and FB models and the capabilities of the robot in terms of visual learning are presented.

In particular, the proposed simulation is inspired by the experiment designed by Liu and collaborators on real flies [15] about visual learning and object recognition. The robot has to explore a square arena, (10m x 10m), in which four objects are present. Even if these objects are different, they can have some similar features. The objects used in this simulation are shown in Fig. 6. Every time the robot meets an object, it tries to recognize that object, extracting features and comparing them with the stored ones. If the robot meets an object for the first time, it extracts and stores the new features. It has been assumed to consider six features: color (in the Hue Saturation Brightness representation, here only the Hue value is considered), orientation, size, center of

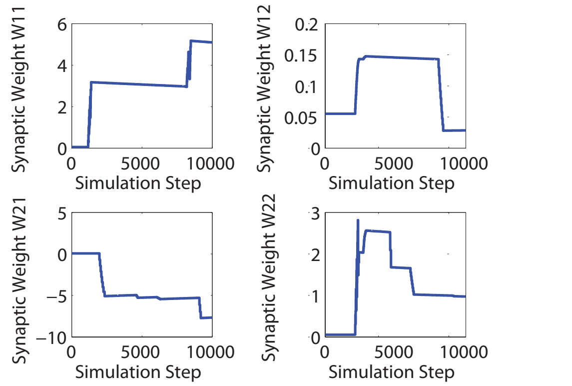

Fig. 5. Simulation results of the olfactory learning model: trend of the synaptic weights during the simulation. The implemented network and the meaning of the parameters are illustrated in Fig. 2.

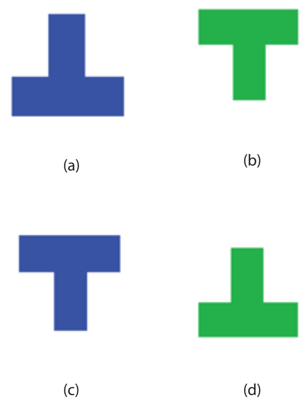

Fig. 6. Objects used in the fan-shaped body model simulation. (a) blue inverted T-shape; (b) green upright T-shape; (c) blue upright T-shape; (d) green inverted T-shape.

gravity position, wideness and height. The meaning of the features is clarified in [10]. The PB model has been set so that the robot is able to detect objects in a range of four meters. Objects triggering punishment shock the robot if its distance from these objects is less than 2.7 meters. In this experiment the color "green" is a bad feature: the robot will be punished every time it tries to approach a green object. Therefore the robot has to learn to avoid green objects. The arena and the simulation results are shown in Fig. 7. At the beginning of the simulation, the robot tries to approach every object standing in its visual range. If punished, the robot increases the *punishment value* of the features of the approached object. If an object is neutral, the punishment value of the features associated to that object decreases. If the escaping value of an object reaches a threshold, the robot will escape when that object is detected. In this simulation the robot learns correctly to avoid green objects. Fig. 8 shows also the punishment value of the bad feature (green color) and the punishment value of a neutral feature, the wideness, that is the same for all the objects. In order to implement a hysteric response, when the punishment value exceeds 2, it is

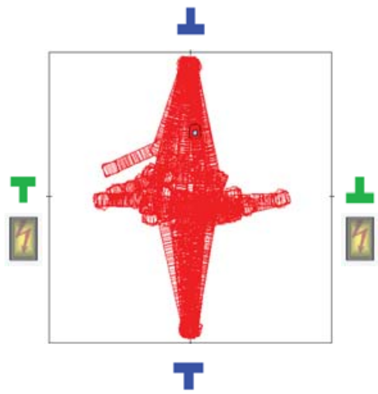

Fig. 7. Robot trajectories obtained during the testing of the fan-shaped body model. After being punished enough times, the robot is able to isolate the dangerous feature (the green color) and escape when a green object is found.

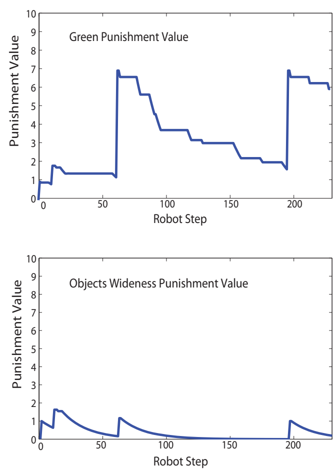

Fig. 8. Comparison between the punishment value of the dangerous feature and the punishment value of a neutral feature, the wideness. Using the punishment value algorithm the robot is able to discriminate the dangerous feature. The decreasing of the Punishing Value of the dangerous feature is due to the steps in which the robot can detect a green object but is not so near to be punished.

simply raised to 7. In this way the robot will remember this bad feature association for a long time, even if the learning is not reinforced. If the robot detects an object, the punishment value of all the features that do not belong to that object will remain the same. A time-dependent decay rate could also be introduced. In a second experiment, the robot has to learn to avoid each “T” object. Color is now neutral for the robot. Even if the shape is not a feature, a T is different from an inverted T because of the different center of gravity. This experiment leads to the same conclusion of the first experiment; the robot is able to recognize bad features and to avoid them.

C. Ellipsoid body

In real fruit flies, the ellipsoid body is necessary for a visual short-term memory and orientation [16]. A polar path integration system is present in the architecture to model

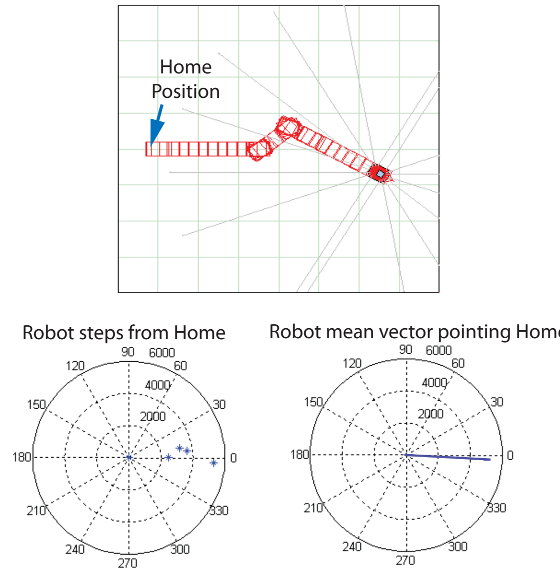

Fig. 9. EB results. The relative position of the robot with respect to the Home is represented in polar coordinates and it is indicated in millimeters (distance) and degrees (angular position). In this case the current robot position is $r = 5730$ mm, $\nu = -12.23$ degrees.

the ellipsoid body. Supposing that an object (in our case the home) is the origin of the polar reference system, the distance of the robot to the object is indicated with r, while the position of the robot is represented by the angle ν.

In the following simulation the behavior of the EB model while the robot is moving around the environment is evaluated. In the simulated environment an odometry error has been introduced, to make the results more realistic. In this first experiment we want to show how the ellipsoid body model works: the robot must be able to orient and update its position while moving into a square arena (8m x 8m). The robot starts from the Home position and moves randomly in the arena: its capability to update its relative position with the Home is analyzed. Of course the coordinates stored into the robot memory will be different from the real ones, because of the odometry errors and the approximation of the path integration method. Fig. 9 shows an example of trajectory and the response of the ellipsoid body. The same test has been repeated many times, in order to make a better analysis of the model.

In order to test the capability of the model in real situations, it is convenient to simulate the robot behavior and the EB response in more complex arenas. In the following experiment the robot has to explore a large arena, in which several objects are present. The robot starts from the Home and initially it moves randomly: this behavior is created to simulate a typical escape reaction of real flies when newly introduced into an arena.

After that, the robot starts an Exploration behavior. If the robot meets objects it is able to learn about their danger or neutrality, thanks to the MBs model. During the exploration, the robot updates its position from the Home. An obsta-

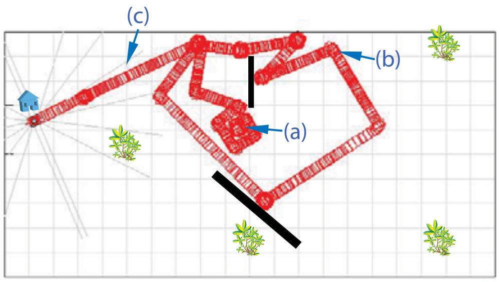

Fig. 10. The robot starts to move and, after the escaping reaction implemented to match the biological experiments with real flies, it begins an exploration (a). The escaping reaction from the Home position to position (a) is not outlined for clarity reasons. After fifteen exploration steps, the battery level is low and the robot starts its homing behavior (b). Using an obstacle avoidance algorithm, the robot is able to return to the Home (c). In order to have a more complex simulation, some objects have been also introduced into the arena, but these did not influence the results.

cle avoidance mechanism was also implemented. During this experiment two behaviors are available: Exploration or Homing. The level of the battery decreases while the robot explores the arena. A virtual battery sensor has been implemented. If the level of the battery is too low, the BSN switches the selected behavior to the Homing behavior. If the stored position is correct, the robot must be able to return to the Home position. Obstacle avoidance is used also during the Homing behavior. Simulation results are shown in Fig. 10. The robot starts to move and, after the escaping reaction implemented to match the biological experiments with real flies, it begins an exploration (a). The escaping reaction from the Home position to position (a) is not outlined for clarity reasons. After fifteen exploration steps, the battery level is low and the robot starts its homing behavior (b). Using an obstacle avoidance algorithm, the robot is able to return to the Home (c).

D. Behavior Selection

In order to allow the robot to choose the “right” behavior, the Behavior Selection Network has been introduced into the model. This network is shown in Fig. 11. The BSN has been tested and its properties have been analyzed. In a real implementation of the model the drives are the inputs of the first layer of the network.

In the following simulations hypothetic drives have been simulated in order to study the response of the BSN in different possible situations. This experiment shows how the Behavior Selection Network works. It has been assumed to have four drives and to represent these drives with four input currents. In this first example the following synaptic weights have been used: $W_{12} = W_{21} = W_{31} = W_{32} = W_{43} = 1.5$; $W_{11} = W_{22} = W_{33} = W_{44} = 10$; $Y_{12} = Y_{13} = Y_{14} = -3$; $Y_{21} = Y_{23} = Y_{24} = -3$; $Y_{31} = Y_{32} = Y_{34} = -3$; $Y_{41} = Y_{42} = Y_{43} = -3$; $Y_{11} = Y_{22} = Y_{33} = Y_{44} = 3$. (see Fig. 11 for the network topology). A random Gaussian noise has been added in the input currents (σ =2). Fig. 12 presents the behavior of the neurons of the network. When a second layer (WTA layer) neuron is firing faster than the

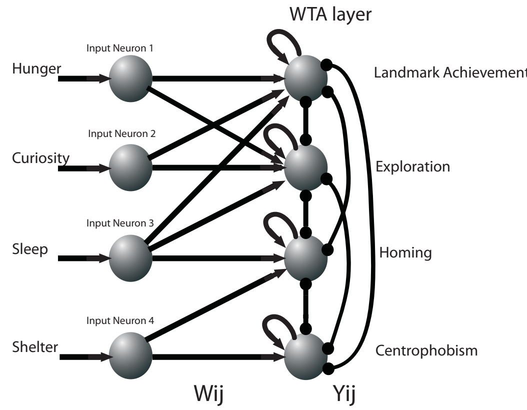

Fig. 11. Spiking network used to simulate the Behavior Selection functionalities. Drives are represented by input currents. Each drive can excite more than one behavior. Synaptic efficiencies between the input layer and the WTA layer represent the influence that each drive has in each behavior. Only the most excited behavior can win the competition and can be selected.

others, the respective behavior is selected. The network has been simulated for ten thousand simulation steps, with an integration step of 20 milliseconds. During a short transient, all the WTA neurons are firing: this situation is due to the response of the synapses between WTA neurons. After this transitory period, only one neuron can win the competition.

V. Real life scenario application

In the previous sections a model of the main parts of a fly brain computational model has been tested. Herewith the capability of the model to solve real useful problems is shown. By modifying the behavior repertoire but maintaining the conceptual structure of the general model we can obtain a versatile robot that is able to learn about the environment, to make choices and to face potentially dangerous situations. The experiment presented in this section is only one example of the real applications of the insect brain model, and it could be easily modified or generalized.

A. Description of the experiment

Let us imagine to have a critical situation in which, after a disaster (e.g. earthquake, fire) it is necessary to rescue people trapped in a place. Often situations like this are very dangerous both for survivors and people who try to help them. Now let us image to have a smart robot able to explore the environment and which can learn, recognize people and remember their position. Such a robot could manage a critical situation acquiring the information needed to solve it. In the present experiment an environment that can represent a place after a disaster has been implemented into a robot simulator. The robot has to explore the environment, find some good objects that it is able to recognize, remember their position and learn about all kinds of danger present in the environment. At the end of the exploration, the robot must escape from the environment and give all the information useful for humans to know the position of the survivors

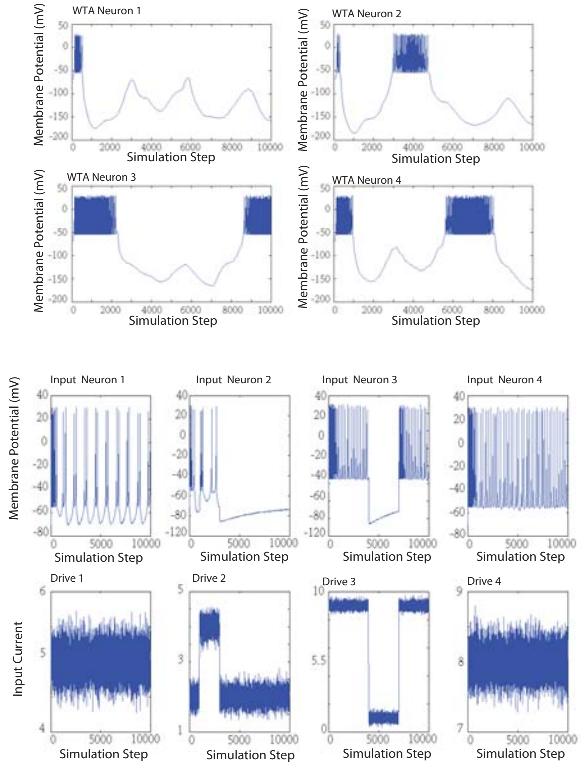

Fig. 12. Results of the simulation of the BSN. After a short transient in which all the WTA neurons are firing, only one neuron can win the competition. The transient is a consequence of the time response of the synapses between WTA neurons. Variation of the drives could also lead to new transient, in which the WTA neurons compete. A low value of the auto-excitatory synapses weights in the WTA layer can cause a continuous switching of the selected behavior, while a too high value leads to a conservative behavior selection.

and organize a safe rescue. In order to solve this problem, the behavior repertoire of the robot has been limited to two possible behaviors, exploration (*rover* type, see [10] for more details) and homing. In the same way, two drives are considered, *Curiosity* and *Sleep*, the latter indispensable for the robot to understand when to leave the environment and return home; for this, a virtual battery level sensor is used. The MBs model was also simplified: only the olfactory learning model will be considered. The synaptic weights, the synapses time constant and the integration step are the same of the previous simulations. To make the simulation light, every robot step of the robot includes only one hundred simulation steps of the MBs and BSN neural networks.

The arena implemented for the simulation and the results are shown in Fig. 13. The Home represents the starting point for the robot exploration and the point the robot has to reach at the end of the simulation. S_1 and S_2 represent the position of the targets: let us assume the robot considers them as interesting objects and, after an approach, it is able to recognize them. Let us assume that the targets are a blue T-shaped object and a blue inverted T-shaped object. Also present are Obj_1, Obj_2 and Obj_3, which are identical among one other. The robot cannot see them, but can sense them thanks to another sensorial system (i.e. olfactory). The robot is punished every time it tries to approach them. In the environment, two other objects are present, a green upright T-shaped object and a green inverted T-shaped object. The robot can detect them with the visual system. The robot is punished only when it tries to approach the first one, while the second one is neutral.

After a long exploration, the robot must be able to detect the targets, learn to avoid as soon as possible the objects Obj_1, Obj_2 and Obj_3, understand that the green upright T-shaped object is dangerous and finally reach the Home and give the position of the targets at the end of the exploration. Mushroom Bodies model will be used for the learning involving Obj_1, Obj_2 and Obj_3; the protocerebral bridge model will be used for the detection of the objects and the fan-shaped body model for the visual learning; the ellipsoid body model is indispensable for homing and remembering the position of the targets. For this simulation, the capabilities of real flies have been extended, for instance, improving the performances of the EB that is now able to store multiple target information in a long term memory. This is an example of how the elementary functions of the *Drosophila* brain that allows the insect to face with its world can be easily extended in a modular way to make a robot able to fulfill more complex tasks, not affordable for the real fly. The Behavior Selection Network is useful to select the homing behavior if the battery level is too low. The parameters of the model have been set so that the robot can sense odors if its distance is lower than three meters away from the nearest odor source, while it is punished if its distance from that source is less than one meter. In the same way, the visual system of the robot can detect objects if they are closer than 2.5 meters. It is punished if an object is closer than 1.5 meters. The arena used for the simulation is 28m long and 15m wide.

B. Results

In this section experimental results from one of the simulations are shown, discussing step by step the behavior of the robot. Only the most relevant robot steps are depicted in Fig. 13, for the sake of clarity. Besides, Fig. 14 shows the MBs model response during the whole simulation.

At step 1 the robot starts the simulation from the Home position. At the second step the robot enters the arena and begins an exploration behavior. The ellipsoid body model updates the position of the robot. Neurons of the MBs model are not stimulated and they lie in their silent state. At the following step (step 5, not shown), the robot uses the increase of the mean free path algorithm. The EB model updates the position of the robot. During the exploration, the robot must find objects and sense odors. At step 9, the robot senses $Odor_1$, but it is not punished, because it is not close enough to Obj_1. In the following step the robot continues its exploration following the increase of the mean free path algorithm, while the EB model updates the position it has stored.

At step 11 (not shown), the robot detects the green T-shaped object. The FB model extracts features from this object and

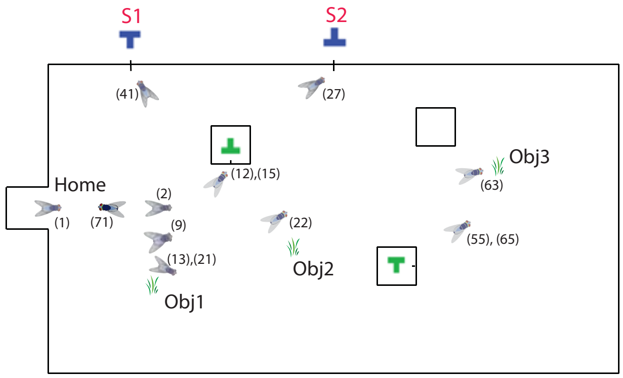

Fig. 13. Most relevant robot steps of the proposed simulation. After the exploration of the environment the robot returns to the Home and gives the position of the target S_1 and S_2. Moreover, information about the dangers in the environment are stored in the FB and the MBs model.

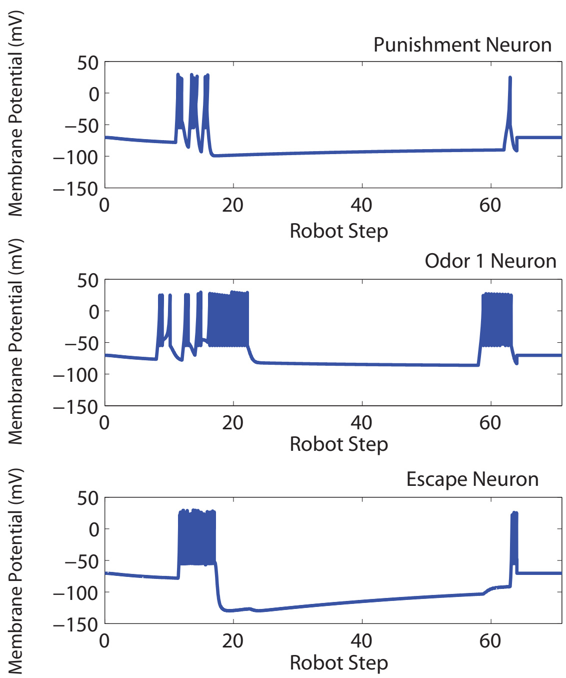

Fig. 14. Mushroom Bodies model response obtained during the simulation.

the robot tries to approach it. While the robot is approaching the new object, it is punished (step 12).

After being punished, the robot escapes from the green upright T-shaped object (step 13). It experiences an unexpected situation: the robot sensed $Odor_1$ but was punished after two subsequent steps, due to the punishing visual input, and not for the odor. So, even if not planned in this way, the robot has made an association between $Odor_1$ and Punishment. This situation is plausible and is a natural consequence of the correlation implementation of the STDP learning. As a consequence, the association between $Odor_1$ and the need to escape is reinforced. While it is escaping, the robot again detects Obj_1, senses $Odor_1$, is punished and escapes again in the opposite direction (step 14, not shown), reaching once more the green inverted T-shaped object (step 15). The robot is then punished for the third time. At step 16 (not shown), the robot is escaping again. At step 20 and 21 the robot is sensing $Odor_1$ again, without being punished. It is very interesting to analyze how the MBs model responds to this contradictory situation. Studying the firing of each neuron of the MBs model, it is possible to see that at a first time the robot was punished immediately after sensing $Odor_1$, while at a second time it senses $Odor_1$ but it is not punished. In this way, at a first time the robot made an association between punishment and $Odor_1$, but at a second time this association was weakened. However, the synaptic weight between the $Odor_1$ neuron and the Escape neuron of the olfactory learning model was not high enough to make the robot escape when sensing again $Odor_1$, without being punished. Now the robot continues its exploration of the arena. At step 22 the robot is near Obj_2, it is sensing $Odor_1$ again but it is not close enough to be punished. The association between punishment and $Odor_1$ must decrease again. While exploring, the robot detects the first target (step 27). The fan-shaped body analogue extracts the features of the object, the robot recognizes the target and tries to approach it. The EB model stored the position of the robot. The target S_2 is now reachable in the future. The robot leaves the object and begins another exploration.

After many steps, the robot detects and reaches the target S_1 and stores its position (step 41). After leaving the second target, the robot begins another long exploration. At step 55, the robot is into the area of detection of the green T-shaped object, but in this case the PB model leads the robot to consider this object repulsive because it is standing in the rear of the robot, therefore the robot leaves the object. The robot continues its exploration and, detecting Obj_3, the robot senses $Odor_1$ again at step 59. At step 63, the robot is close enough to Obj_3 to be punished. Because of the position of the robot, the punishment is not so strong, but the robot is sensing $Odor_1$ and it is recalling the association with punishment: even if the Punishment neuron only spikes once, the robot escapes.

Analyzing MBs response and the synaptic weights at step 64, it is evident how the robot reinforced the association between $Odor_1$ and Punishment, as shown in Fig. 15. Learning allowed the robot to escape fast, without strong punishment. After escaping, at step 65, the robot meets again the green T-shaped object. While the robot tries to approach it, the low output of the virtual battery sensor determines the behavior and initiates homing behavior. The EB model is involved to remember the Home position. The response of the EB model at step 65 is shown in Fig. 16. At steps 67, 69 (not shown) and 71 the robot tries to return to the Home position. At the end of the simulation, the robot can communicate the approximated position of the targets. Moreover, the robot is aware of the association between an odor and a danger. Nevertheless, in this simulation, the robot was not able to safely associate a visual feature with reward or punishment, because it has been punished only once while approaching a landmark.

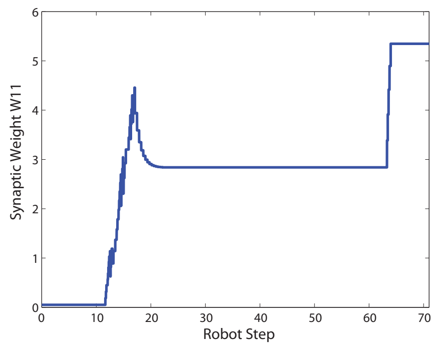

Fig. 15. Trend of the synaptic weight of the synapses between the $Odor_1$ receptor neuron and the Escape neuron, in the pre-motor area. The synaptic weights of the MBs olfactory learning model are subject to STDP learning. The higher the value of the weight is, the faster the robot will escape if punished while sensing that odor. If the weight exceeds a certain threshold, the robot sensing that odor will escape even if not punished at all. For clarifications about the parameters, the model implemented is shown in Fig. 2.

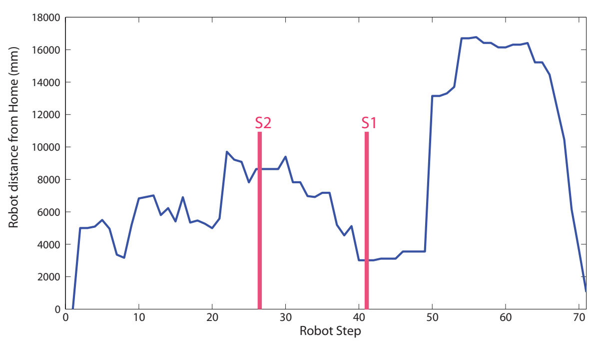

Fig. 16. During the simulation, the EB model estimates the distance of the robot from the Home position. Errors in the position are due to the simulated odometry error and to the path integration method approximations.

VI. Remarks and Conclusions

A computational model inspired by the *Drosophila melanogaster* brain has been implemented within a robot simulator.

The olfactory learning model is a two layer spiking neural network and the STDP algorithm has been used for the learning implementation. This algorithm allows the robot to make associations involving odors in complex environments. The orientation memory has been tested through the simulation of the ellipsoid body model. The Behavior Selection Network has been simulated with hypothetic drives. Moreover, the whole architecture has been simulated in an application useful for real life scenarios. The reported results demonstrate that the proposed computational model can be successfully applied to real robots to solve tasks in complex environments.

Acknowledgement

This work was supported by EU Project SPARK II, Spatial-temporal Patterns for Action-oriented perception in Roving Robots II. The authors acknowledge the constant support of Prof. R. Strauss on the biological aspects that helped in the assessment of the simulation results.

References

[1] B. Webb, “Can robots make good models of biological behaviour?”, *Behavioral and Brain Sciences*, vol. 24, pp. 1033-1050, 2001.

[2] R.D. Beer, R.D. Quinn, H.J. Chiel, R.E. Ritzmann, “Biologically-inspired approaches to robotics”, *Communications of the ACM*, vol. 40, pp. 30-38, 1997.

[3] N. Franceschini, J.M. Pichon, C. Blanes, “From insect vision to robot vision”, *Phil. Trans. R. Soc. Lond. B.* , vol. 337, pp. 283-294, 1992.

[4] B. Webb, “Using robots to model animals: A cricket test”, *Robots and Autonomous Systems*, vol. 16, pp. 117-134, 1995.

[5] G. Hartmann, R. Wehner, “The ant’s path integration system: a neural architecture”. *Biol. Cybern.*, vol. 73, pp. 483-497, 1995.

[6] D. Lambrinos, R. Moller, T. Labhart, R. Pfeifer, R. Wehner, “A mobile robot employing insect strategies for navigation”, *Robotics and Autonomous Systems*, vol. 30, n 1, pp. 39-64, 2000.

[7] M. Mronz, R. Strauss, “Visual motion integration controls attractiveness of objects in walking flies and a mobile robot”, *International Conference on Intelligent Robots and Systems, Nice*, pp. 3559-3564, 2008.

[8] L. Alba, P. Arena, S. De Fiore, L. Patané, R. Strauss, G. Vagliasindi, “Implementation of a Drosophila-inspired orientation model on the Eye-Ris platform”, to appear in proceedings of Int. Conf. on Cellular Non linear Networks and their Applications, 2010.

[9] M. B. Reiser, M. H. Dickinson, “A test bed for insect-inspired robotic control”, *Phil. Trans. R. Soc. Lond.*, vol. 361, pp. 2267-2285, 2003.

[10] P. Arena, C. Berg, L. Patané, R. Strauss, P.S. Termini, “An insect brain computational model inspired by *Drosophila melanogaster*: architecture description”, submitted to *International Joint Conference on Neural Networks*, 2010.

[11] K.G. Götz, R. Biesinger,“Centrophobism in Drosophila melanogaster. Physiological approach to search and search control”, *J. Comp. Physiol. A*, vol. 156, pp. 329-337, 1987.

[12] J.S. de Belle, M. Heisenberg M, “Associative odor learning in *Drosophila* abolished by chemical ablation of Mushroom Bodies” *Science*, vol.263, pp.692-695, 1994.

[13] X. Liu, R.L. Davis, “Insect olfactory memory in time and space”, *Curr. Opin. Neurobiol.*, vol. 6, pp. 679-685, 2006.

[14] C. Margulies, T. Tully, J.Dubnau, “Deconstructing memory in *Drosophila*”, *Curr. Biol.*, vol.15, pp.700-713, 2005.

[15] G. Liu, H. Seiler, A. Wen, T. Zars, K. Ito, R. Wolf, M. Heisenberg, L. Liu, “Distinct memory traces for two visual features in the Drosophila brain”, *Nature*,vol. 439, pp. 551-556, 2 February 2006.

[16] K. Neuser, T. Triphan, M. Mronz, B. Poeck, R. Strauss, “Analysis of a spatial orientation memory in *Drosophila*”, *Nature*, vol. 453, pp. 1244-1247, 2008.

[17] U. Hanesch , K.F. Fischbach, M. Heisenberg, “Neuronal architecture of the Central Complex in *Drosophila melanogaster*” *Cell Tissue Res.*, vol.257, pp.343-366, 1989.

[18] R. Strauss, “The Central Complex and the genetic dissection of locomotor behaviour”, *Curr. Opin. Neurobiol.*, vol. 12, pp. 633-638, 2002.

[19] M. Besson, J.R. Martin, “Centrophobism/thigmotaxis, a new role for the mushroom bodies in Drosophila” *J. Neurobiol.* 62: 386396, 2005.

[20] S. Song, K.D. Miller, L.F. Abbott, “Competitive Hebbian learning through spike-timing-dependent synaptic plasticity”, *Nat Neurosci* vol.3, pp.919-926, 2000.

[21] S. Song, L.F. Abbott, “Cortical development and remapping through Spike Timing-Dependent Plasticity”, *Neuron*, vol. 32, pp. 339-350, 2001.

[22] M. Mronz, “Die visuell motivierte Objektwahl laufender Taufliegen (*Drosophila melanogaster*). - Verhaltensphysiologie, Modellbildung und Implementierung in einem Roboter. *PhD thesis*, University of Wuerzburg, 2004.

[23] E.M. Izhikevich, “Simple Model of Spiking Neurons”, *IEEE Transactions on Neural Networks*, vol. 14, n6, pp. 1569-1572, 2003.