Insect inspired unsupervised learning for tactic and phobic behavior enhancement in a hybrid robot

Paolo Arena, Sebastiano De Fiore, Luca Patané, Massimo Pollino and Cristina Ventura

***Abstract*— In this paper the implementation of a correlation-based navigation algorithm, based on an unsupervised learning paradigm for spiking neural networks, called Spike Timing Dependent Plasticity (STDP), is presented. The main characteristic of the learning technique implemented is that it allows the robot to learn high-level sensor features, based on a set of basic reflexes, depending on some low-level sensor inputs. The goal is to allow the robot to autonomously learn how to navigate in an unknown environment, avoiding obstacles and heading toward or avoiding the targets (on the basis of the rewarded action).**

This algorithm was implemented on a bio-inspired hybrid mini-robot, called TriBot. The peculiar characteristic of this robot is its mechanical structure, since it allows to join the advantages both of legs and wheels. In addition, it is equipped with a manipulator that allows to add new capabilities, like carry objects and overcome obstacles.

Robot experiments are reported to demonstrate the potentiality and the effectiveness of the approach.

I. Introduction

It is desirable that robots would be, as much as possible, autonomous and self-sufficient; this requires a control algorithm that allows the robot to navigate through the environment on the basis of the information acquired from it. Since the environment is unknown and in continuous evolution, it is not possible to a priori program the behavior of the robot. Therefore, the control algorithm has to be dynamic in order to allow the robot to change its behavior depending on the situation. The idea is, in fact, not to create, at least at this stage, an internal representation of the environment, but to learn a general method that allows the robot to reach specific objects, that represent the target, interacting continuously with the environment. At this aim, it is necessary that the robot is equipped with different kinds of sensors, like contact, vision and hearing sensors.

To implement robust and effective control algorithms, neural networks are often used [1]. These ones have seen an outburst of attention over the last few years and are being successfully applied across a notable range of problem domains, in areas as diverse as finance, physics, engineering, geology and medicine. They allow, in fact, to process information in a living-being-like manner, i.e. learning from the experience. Interesting information for the network design process can be gathered from the insect world. In insects, the basic locomotion abilities are generated typically within the thoracic ganglia, while higher parts of the brain have the different role to modulate the basic behaviors to give rise to more complex capabilities. For example, the insect Mushroom Bodies (MBs) neural structure is thought to be responsible for anticipation abilities [2]. In particular, a function ascribed to MBs consists in the enhancement of causal relations arising among the insect basic behaviors, by exploiting the temporal correlation between sensory events. Information storage and retrieval in the case of the olfaction sense are demonstrated in [2], [3]. Another interesting element, that can be used for the development of a real time control architecture for autonomous robots, is the biological mechanism of spike-timing-dependent plasticity (STDP). This learning algorithm, initially is a suitable learning paradigm for spiking networks and can be used to model synaptic plasticity. In particular, in this application, a Spiking Neural Network (SNN) has been used to implement the navigation control algorithm for TriBot, the bio-inspired hybrid robot used as a test-bed [4]. Spiking neural networks fall into the third generation of neural network models, which explicitly take into account the timing of inputs, increasing the level of realism in a neural simulation. The network input and output are usually represented as series of spikes. SNNs have an advantage of being able to process information in the time domain [5]. Moreover, essential features of neurons and their interconnections can be easily programmed into a computer, which then simulates the brain's learning processes. Starting from the experience, that can be a priori given or can be built during time, with or without human supervision, a neural network is therefore able to recognize images, sounds, shapes and so on.

Associative learning is a basic principle in nature. In the proposed application, on the basis of recent results on the olfactory learning process in the MBs acquired from experiments in the Drosophila melanogaster [6], the learning process is inspired by the Classical Conditioning paradigm, which is a reflexive or automatic kind of learning in which an initially neutral stimulus acquires the capacity to evoke a response that was originally evoked by another stimulus [7].

The original model of Classical Conditioning was most famously demonstrated by Ivan Pavlov [8]. It begins with the observation that certain stimuli, referred to as unconditioned stimuli (US), reliably yield an unconditioned response (UR). When a neutral stimulus is paired with the US, it may also yield the same response through conditioning. Under these conditions the neutral stimulus is referred to as the conditioned stimulus and the response to the CS is the conditioned response (CR). One important characteristic of the Classical Conditioning is that, if the conditioned and unconditioned stimuli are not paired for a given number of trials an organism will stop exhibiting the conditioned response.

Similarly to what happens in nature, connections between

Paolo Arena, Sebastiano De Fiore, Luca Patané Cristina Ventura are with the Dipartimento di Ingegneria Elettrica, Elettronica e dei Sistemi, Universitá degli Studi di Catania, Italy (email: {parena,lpatane}@diees.unict.it).

neurons are not a priori fixed, but they change during the experiment, getting stronger or weaker on the basis of the stimuli coming from the external environment. This characteristic is the base of the learning algorithm used in this application. Synaptic plasticity plays, in fact, central roles for memory, learning, and the development of the brain. More precisely, a synapse is strengthened if the postsynaptic spike follows the presynaptic spike. This synaptic modification is called long-term potentiation. On the other hand, a synapse is weakened if the presynaptic spike follows the postsynaptic spike, which is called the long term depression [9].

In literature, various architectures, based on spiking neural networks, have been implemented to control the navigation of mobile robots [10], [11], [12].

In [10] the relation between neural dynamics and robot behavior is studied. The aim there was to develop self-organizing algorithm of spiking neural networks, based on genetic algorithms, applicable to autonomous robots. In a similar way, in [11] a novel mechanism for controlling autonomous mobile robots that is based on spiking neural networks (SNNs) is introduced. Also in this case an adaptive genetic algorithm (GA) is used to evolve the weights of the SNNs online using real robots. They demonstrate that the adaptive GA, using adaptive crossover and mutation probabilities, converges in a relatively short time interval in a small number of generations and produced a good solution that outperformed the standard GA. The evolved SNN controller also provided an acceptable solution to the wall following problem even when compared with a Fuzzy controller benchmark; the SNN controller gave a smoother and better response.

Once again, in [12] a spiking neural network (SNN) is used for behavior learning of a mobile robot in a dynamic environment including multiple moving obstacles, but in this case the learning method of SNN uses outputs from fuzzy controllers as teaching signals.

In this paper, the learning technique implemented is inspired by the Classical Conditioning and it is realized using a Spike Timing Dependent Plasticity (STDP) algorithm. Our aim is to demonstrate how the robot is able to learn to recognize and, initially, to approach all the targets present in the environment, which are represented by yellow and blue circles randomly placed on the ground, and then to learn to avoid, for example, the blue ones, depending on the robot needs.

The robot used for the experimental results is a modular hybrid robot named TriBot. The TriBot structure is constituted by two wheel-legs modules, an optimal solution for walking in rough terrains and to overcome obstacles, inspired by wheel-legs robots like Prolero, Asguard, RHex and Whegs [13]. Moreover, a manipulator was added to improve the capabilities of the system that is able to perform various tasks: environment manipulation, object grasping, obstacle climbing and others.

The paper is organized as follows: the learning technique and the STDP rules are initially described and the main characteristics of the TriBot robot are then presented. The structure of the neural architecture implemented in this application is analyzed in Section IV and finally, in Section V, experimental results show as the implementation of an unsupervised learning structure, like STDP, allows the robot to learn how to autonomously navigate through the environment approaching or avoiding targets.

II. Spiking neurons and STDP rule

In this section the mathematical model used to simulate the spiking neuron behavior and the STDP rule applied to implement the learning algorithm are presented.

There exist different neural dynamical properties of biological neurons, such as spiking behaviors (tonic, phasic, and chaotic spiking) and bursting behavior [14]. All of these behaviors can be simulated using the neuron model proposed by Izhikevich. This mathematical model is based on a system of two first order differential equations [15]:

$$\begin{aligned} \dot{v} &= 0.04v^2 + 5v + 140 - u + I \\ \dot{u} &= a(bv - u) \end{aligned} \tag{1}$$

with the auxiliary after-spike resetting:

$$\texttt{if } v \geq 0.03, \texttt{ then} \begin{cases} v \leftarrow c \\ u \leftarrow u + d \end{cases} \tag{2}$$

where v and u are dimensionless variables, and a, b, c and d are dimensionless parameters. The variable v represents the membrane potential of the neuron and u represents a membrane recovery variable. Synaptic currents or injected dc-currents are delivered via the variable I. The time unit is ms.

The various behaviors of the neuron are obtainable varying the parameters a, b, c and d of the mathematical model.

In this application, we have chosen for all neurons the class I excitable model, whose main characteristic is that the spiking rate is proportional to the amplitude of the stimulus. This characteristic allows to encode the strength of the input into firing rate of the neurons. Such property is really important since it gives the possibility to fuse sensory data at the network input level. Here we have fixed $a = 0.02$, $b = -0.1$, $c = -55$ and $d = 6$ (i. e. class I excitable neurons [14]), whereas the input I accounts for both external stimuli (e.g. sensorial stimuli) and synaptic inputs.

Neurons are, then, connected through synapses. Considering a neuron j which is connected with n neurons, and indicating with t_s the instant in which a generic neuron i, connected to neuron j, emits a spike, the following equation represents the synaptic input to neuron j:

$$I_j(t) = \sum w_{ij}\varepsilon(t - t_s) \tag{3}$$

where w_{ij} represents the weight of the synapse from neuron i to neuron j, while the function $\varepsilon(t)$ is given by the following formula:

$$\varepsilon(t) = \begin{cases} \frac{t}{\tau}e^{1-\frac{t}{\tau}} & \texttt{if } t \geq 0 \\ 0 & \texttt{if } t < 0 \end{cases} \tag{4}$$

Equation (4) describes the contribution of a spike, from a presynaptic neuron emitted at $t = 0$. In this application τ has been fixed to $\tau = 5ms$.

The Spike Timing Dependent Plasticity (STDP) [16] is a bio-inspired learning technique which realizes an unsupervised Hebbian learning scheme that modifies synaptic weights according to their timing between pre- and post-synaptic spikes. Therefore, weights changes according to STDP rules [17]:

$$\Delta w = \begin{cases} A_+ e^{\frac{\Delta t}{\tau_+}} & \texttt{if } \Delta t < 0 \\ A_- e^{\frac{-\Delta t}{\tau_-}} & \texttt{if } \Delta t \geq 0 \end{cases} \tag{5}$$

where $\Delta t = t_{pre} - t_{post}$ is the temporal difference between the pre (t_{pre}) and post (t_{post})-synaptic spikes. If $\Delta t < 0$ and so the post-synaptic spike occurs after the pre-synaptic spike, thus the synapsis should be reinforced. Otherwise, if $\Delta t \geq 0$ (the post-synaptic spike occurs before the pre-synaptic spike), the synaptic weight is decreased by the quantity Δw. The choice of the other parameters (A_+, A_-, τ_+ and τ_-) of the learning algorithm will be discussed below. The term A_+ (A_-) represents the maximum Δw which is obtained for almost equal pre- and post-spiking times in the case of potentiation (depression). While, the parameters τ_+ and τ_- determine the ranges of pre-to-post synaptic inter-spike intervals over which synaptic strengthening and weakening occur.

The use of the synaptic rule described by equation (4) may lead to an unrealistic growth of the synaptic weights. Synapses, in fact, are functions of the time difference between pre- and post-synaptic spikes only, independently of the synaptic weight [17]. This no-weight-dependent STDP requires imposing upper and lower bounds on the weights to prevent unlimited weight growth [18], [19].

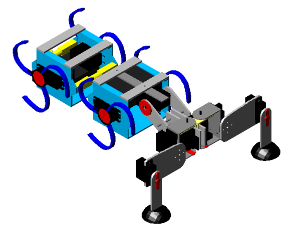
An additional problem also arises: multiple inputs can activate different synapses of a neuron and when all of them get saturated at their upper bounds, this neuron will not have any discrimination ability. This problem can be solved by introducing constraints such as limiting the total synaptic strength of a neuron $\sum_i w_{i,j}^2 = const$ [20]. For the excitatory connections, we use a quadratic normalization rule:

$$w_{i,j}(t) = W \cdot w_{i,j}(t - dt) / \sqrt{\sum_{i=0}^{I-1} w_{i,j}^2(t - dt)} \tag{6}$$

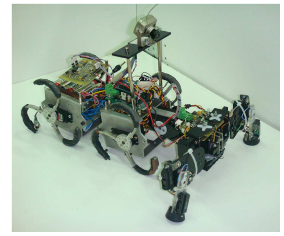
where I is the number of the first layer neurons, while W is a constant value used for normalization.

Furthermore, some authors (see for instance [21], [22]) introduce a decay rate in the weight update rule. This solution avoids that the weights of the network increase steadily with training and allows a continuous learning to be implemented. In the simulations, the decay rate has been fixed to the 0.1% of the weight value and is performed each step. Thus, the weight values of all plastic synapses are updated according to the following equation:

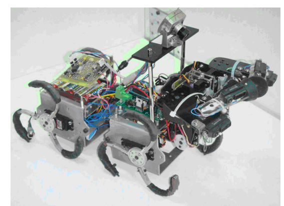
$$w_{i,j}(t + dt) = 0.999 w_{i,j}(t) + \Delta w_{i,j} \tag{7}$$



(a) AutoCAD design;



(b) Manipulator down;



(c) Manipulator up;

Fig. 1. Hybrid robot TriBot.

The presented learning mechanisms allow to increase the adaptation capabilities of the network to dynamically changing environment as it will be shown in the robot experiments.

III. TriBot Robot

In this section, we briefly discuss about the mechanical and electronic characteristics of TriBot, the autonomous mobile hybrid robot used during the experiments.

The mechanical design of the robotic structure and the first prototype are shown in Fig.1.

The robot has a modular structure, in particular it consists of two wheel-legs modules and a two-arms manipulator. The two wheel-legs modules where chosen since they are an optimal solution for walking in rough terrains and to overcome obstacles.

In fact, each leg of the robot TriBot has a peculiar shape, it is an hybrid solution, the result of a study on the efficiency of a wheel-leg hybrid structure; in this way the robot can have the advantages of using legs, easily overcoming obstacles and facing with rough terrains. On the other way, wheel-legs have the shape of wheels, therefore the robot TriBot is able to have a quite smooth movement in regular terrains and so to reach high speed.

The two wheel-legs modules are connected using a passive joint with a spring that allows only the pitching movement. This joint facilitates the robot during climbing, in fact the body flexion easily allows, in a passive way, to adapt the robot posture to obstacles.

Moreover, a manipulator, that consists of two legs/arms with three degrees of freedom, was added to improve the capabilities of the system that is able to perform various tasks: environment manipulation, object grasping, obstacle climbing and others. This manipulator is connected to the

TABLE I
TECHNICAL CHARACTERISTICS OF TRIBOT.

Weigh [Kg]	1,95
Dimensions [cm] (length x height x width) (manipulator up)	36x23x13
Dimensions [cm] (length x height x width) (manipulator down)	28x12x25
Velocity [cm/s]	46
Wheel-legs motors	5x Hitec HS-985MG
Manipulator motors	6x Hitec HS-82MG
Motors batteries	10 x 3000 mAh@1.2V stylus AA
Control Board battery	1600 mAh@11.1V Lithium polymers
Higher obstacle overcoming (using only wheels)	1.42 times wheel radius
Higher obstacle overcoming	1.8 times wheel radius

central wheel-legs module through an actuated joint, which allows the robot to assume two different configurations, therefore the manipulator can be useful to improve locomotion capabilities when it is moved down Fig.1(b) (i.e. used as legs), whereas, when it is moved up it can make manipulation and grasp objects Fig.1(c) (i.e. used as arms).

The main technical and mechanical characteristics of the robot are shown in Table I.

To control the robot, two boards based on ATmega128, a low-power CMOS 8-bit microcontroller architecture connected using a serial bus and a graphical user interface (GUI) have been developed. Besides, the computer manages and controls the robot behavior through a RF wireless XBee module, that uses the standard ZigBee protocol.

The main role of the master control board is to control the servomotors that actuate the four wheel-legs and is also used as a bridge between the PC and the other board mounted on the manipulator. The manipulator is controlled by a similar board, configured as slave, which is used to give the PWM signals to the six servomotors that actuate the manipulator and to the servomotor that actuates the joint connecting the manipulator with the rest of the robot. This board has also the important task to read data from the distributed sensory system embedded in the manipulator. In fact, to use the robot as test bed for perceptual algorithms, a sensory system is needed. In particular, on the manipulator, four distance sensors have been distributed for obstacle detection in order to make the system able to safely move in unknown environments and a series of micro-switches are used to detect collisions and to grasp objects.

Moreover, the robot is equipped with a wireless camera that can be used for landmark identification, following moving objects, and other higher level tasks.

IV. THE SPIKING NETWORK

To realize the control algorithm, a spiking neural architecture (Fig.2) has been used, that allows the robot to avoid obstacles and to recognize and reach targets. The architecture

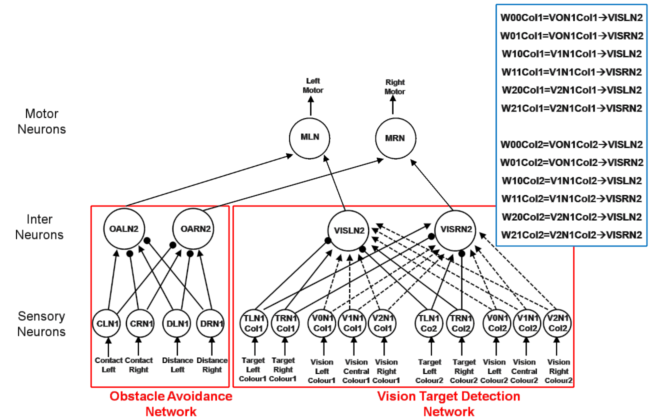


Fig. 2. The whole neural architecture, composed by two subnetwork: Obstacle Avoidance Network and Vision Target Detection Network. The inputs of the first subnetwork are: Contact Left, Contact Right, Distance Left and Distance Right that represent the signals coming from the contact and distance sensors. Respectively, the inputs of the Vision Target Detection Network are: Target Left Colour1 and Target Right Colour1 that represent signals coming from the target sensors (photo-resistances) and Vision Left Colour1, Vision Central Colour1, Vision Right Colour1 that represent signals coming from the vision sensor that analyzes the scene dividing the image into three vertical visual areas. Solid-lines indicate fixed synapses, while dashed-lines are used for synapses subject to learning. They are replicated for the Colour2.

here proposed is composed by three layers constituted by: sensory-neurons, inter-neurons and motor-neurons.

In this case, the unconditioned stimuli (US) that cause an unconditioned response (UR) are the contact and the target sensors, whereas the conditioned stimuli (CS) are the distance and vision sensors. As previously said, the response UR to the US is a priori known, that means that the synaptic weights that connect US and UR do not change during the simulation. On the contrary, the synaptic weights that connect CS and CR can change during the experiment according to the STDP rules previous described. They are initialized to the same value, selected such that, at the beginning of the simulation, no response is associated to these stimuli. Mainly, the neural network here proposed, can be divided in two main blocks, each of them deals with a different aspect of the control algorithm: Obstacle Avoidance Network and Vision Target Detection Network.

Both networks are then connected to the motor-neurons which control the robot movement. These neurons, in fact, represent the last layer of the network and they give the input to the motors of the robot, in particular, if the right motor-neuron emits more spikes than the left one, the robot will turn on the left, and vice versa.

The first block of the network has the task of managing the interaction between the robot and the environment. The inputs are signals coming from the contact sensors, that represent the unconditioned stimuli and from the distance sensors that, instead, represent the conditioned stimuli; here contact sensors are simulated through a suitable threshold on the signals coming from the same distance sensors. As shown in Fig.2, all synaptic weights of this block of the network have been a priori fixed. That means that the robot is a priori able to avoid obstacles. The functional principle is the following: if, for example, a left contact is detected, the left motor-neuron emits more spikes than the right one and consequently the robot turns right. Moreover, distance sensors work in the same way as contact sensors, but they allow to avoid that the robot bumps against obstacles giving the network, in advance, the right inputs in order to turn on the correct side. Since the rotation angle is proportional to the spikes emitted by the motor-neurons, it will be proportional to the distance between the robot and the obstacles. This network here considered, already available as a basic behavior for the robot, can also be learned autonomously as discussed in [5].

Furthermore, the Vision Target Detection Network allows the robot to visually recognize and reach targets. The neurons of the sensory layer of this block are five per each detected visual feature (in this case the color). In fact, in Fig.2, in the sensory layer there are two sets of five neurons because there are two types of targets to be detected (other types of targets can be easily added). Also this block of the network can be divided in two subnetworks: Target Detection and Target Vision. The first one has as inputs the signals coming from the target sensors (photo-resistances) and it allows the robot to align itself with the target, when it is detected. As shown in Fig.2, the synaptic weights connecting neurons of this subnetwork are fixed (solid-lines), in fact, they represent the unconditioned stimuli.

The Vision Target subnetwork is constituted of three neurons in the sensory layer and two in the inter layer which are shared with the Target Detection Network. The input are signals coming from the visual sensor. Synaptic weights connecting neurons of this subnetwork are variable (dashed-lines) since the vision sensor represents for the robot the conditioned stimulus. Practically, the image captured by the camera is divided in three vertical sections, therefore it is possible to individuate if there is a target and if it is on the left, right or in front of the robot. If the vision sensor detects a target in a precise area of the image, the correspondent neuron is excited and consequentially, it emits spikes. Since the synaptic weights that connect the sensory and the inter layers of this subnetwork are initialized to the same value, the robot has no response to this stimulus. But, if a target is detected after that a neuron of the sensory layer has emitted spikes, i.e. after a US neuron was excited, the implemented learning algorithm reinforces the synaptic weight connecting these two neurons according to STDP rules previously described, while the other connections are weakened. When, after a number of similar stimuli-response associations, this synaptic weight reaches a value that allows to excite the inter-neurons, the robot will be able to head itself toward the target before detecting it through the target sensors.

In this application two kinds of targets have been used. The aim is to demonstrate that, using a neural network based on the STDP learning technique, the robot initially learns to approach both targets and then, it can learn to avoid one of them depending on the changes in the reward response obtained through the target sensors. This is also common in nature: insects can learn to associate specific odors to food, but they are also able to modify this association on the basis of the environmental changes. In our experiment, at the beginning both targets are rewarding and therefore the robot learns to reach them; after this, the robot begins to be punished whenever it reaches one of the target (i.e. the target represented by the blue circle), consequently the robot learns to avoid it having a repulsive behavior, forgetting, at the same time, the previously acquired attractiveness. The forgetting capability is a natural consequence of the decay rate implemented into the STDP learning strategy: this allows to cope with environmentally changing conditions, like the one described in this example. In a plausible scenario, a robot could find a charging station and use it for some time. After that, the station could no longer be able to recharge the robot (for example for a failure). In this case the robot should be able, in a few trials, to detect the new condition, forget the previously acquired association, in search of a new, rewarding one. The strategy is implemented here by changing the robot association to that specific target, at the beginning, in fact, the robot is rewarded when it finds a blue circle, whereas in this new condition it is punished. This is realized changing the sign to the synapses associated to the unconditioned stimuli in the network.

Fig. 3. Environment used during the experiments. The arena has been filled with 3 yellow plus 3 blue targets. An obstacle has been added to demonstrate that the robot structure guarantees the effectiveness of the control strategy also in more complex scenarios where climbing actions have to be performed

In a strict biological perspective, in the insect Mushroom Bodies two different pathways were found: one appetitive, octopamine mediated, and one aversive dopamine dependent [6]. As a consequence, these two paths are elicited as a function of the external reward or shock signals. In our implementation, we purposely simplified the structure considering only one internal synaptic path connecting the target sensory layer to the inter-neurons, and we enabled a switching between aversive and appetitive unconditioned stimuli by simply changing the sign of the fixed synapses between target and inter-neurons.

V. Experimental Results

The robot experiments have been performed in the arena shown in Fig.3, with dimension 3mx2,2m, randomly filled with yellow and blue circles that, for the robot, represent the targets.

During the learning phase the robot navigates in the environment using only the reflex-based, inherited behaviors that rely on contact sensors and short range target detector sensors.

The learning procedure consists of several trials in which the synapses subject to the STDP learning rule are updated. In fact, while the robot navigates through the arena, it may meet a yellow or blue circle in the cone of vision of the camera. It then chooses the circle whose area occupies the most of the visual scene. If the identified object is yellow or blue and, at the next control step, the robot detects a target, the corresponding neurons are excited and consequently the synapse that connects them is reinforced, while the others are weakened. All plastic weights are initialized to the same value: $w_{i,j} = 3.464$ in order that $\sqrt{\sum_{i=0}^{I-1} w_{i,j}^2(t-dt)} = W$ where $W = 6$ is the normalization factor. The learning parameters chosen are $A_+ = A_- = 0.05$.

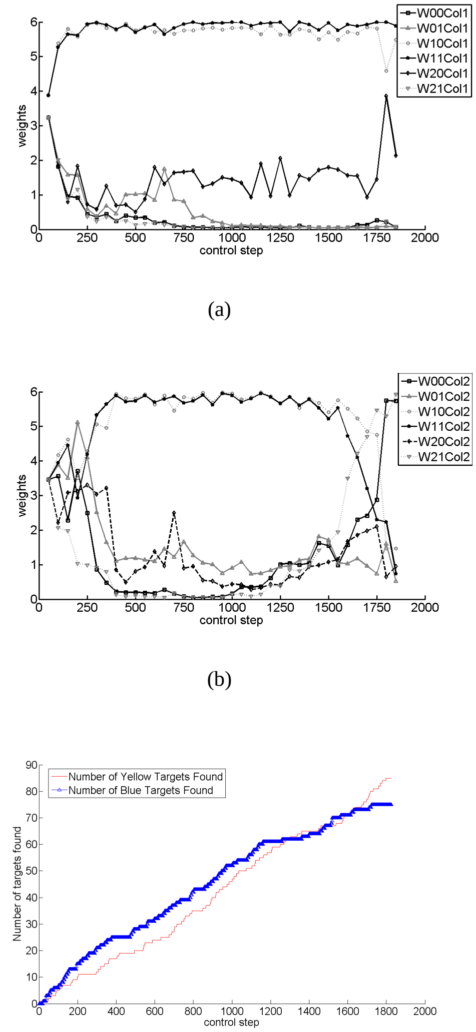
Fig.4 shows the trend of the number of yellow and blue targets found by the robot and the correspondent learnable synaptic behaviors.

A control step corresponds to a robot action that is obtained on the basis of the spikes emitted by the motor-neurons. For each control step the network elaborates the sensory inputs for 333ms.

In the first part of Fig.4(a) and Fig.4(b) the evolution of the synapse values demonstrates how the robot learns to correctly interpret the preprocessed visual input. In fact, the rewarding information coming from the target sensors allows the creation of the anticipative action of the targeting when one of the two targets is visible in the scene. This is evident in Fig.5, where the trajectories followed by the robot at different times during the experiment are reported.

At the beginning of the experiment the robot navigates in the environment driven by the obstacle avoidance network (Fig.5(a)). In this case, when a target is found, by chance (i.e the robot passes through a colored circle on the ground), the learning process starts to modify the network dynamics. After about 30 minutes of running, the robot improved its navigation skills, orienting its trajectories toward both yellow and blue targets, detected through the camera (Fig.5(b)). After 40 minutes the US coming from the target sensor associated to the blue targets change from rewarding to punishing. In terms of the network structure in Fig.2 the change corresponds to a modification of the association between the target left and right Colour2 inputs and the corresponding inter-neurons. The two inputs are exchanged: this corresponds to consider the stimulus as aversive instead of attractive. The effect of the new environmental condition is perceived by the network that changes its dynamics. The learning of the new condition is visible in Fig.4(b) where the synapsis related to the blue targets begin to change (after control step 800) producing a modification of the robot behavior, as shown in Fig.5(c). The robot, after 30 minutes from this change, is able to collect only the yellow targets avoiding the blue ones that are no more rewarding. The performance of the system are also reported in Fig.4(c), where the cumulative number of targets found is reported. At the beginning of the experiment the number of retrievals increases for both targets, while in the second part of the experiment the further number of retrieved blue targets starts to decrease since the robot have learned to avoid them. Moreover, thanks to the hybrid structure of the system, the learned behavior can be applied also to more complex environment, for instance where obstacles are included (see Fig.5(c)). In this situation, the robot can associate to the recognized obstacle a climbing action, that is performed using the frontal manipulator as additional legs, see [25] for more details.

The obtained results show that the robot is able to learn correlation between visual sensory stimuli and target sensors stimuli in various circumstances, allowing to cope with dynamically changing environment, exploiting the dynamic

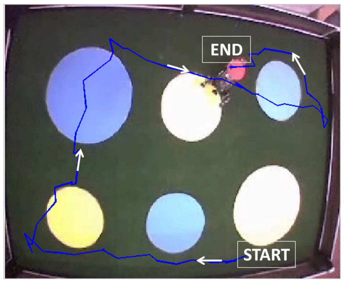


(a)

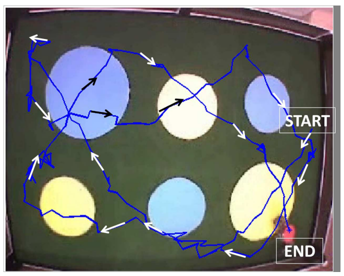
(b)

(c)

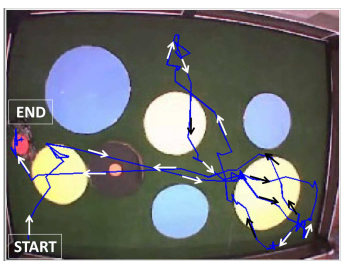
Fig. 4. Evolution of the network during the experiment; (a) Learnable weights relative to the yellow target; (b) learnable weights related to the blue target. At control step 800 the rewarded action associated to the blue target is changed and therefore the robot starts to learn to avoid it; (c) Cumulative number of yellow and blue targets found.



(a)



(b)



(c)

Fig. 5. Trajectory followed by the robot captured through a camera placed on the roof: (a) trajectory at the beginning of the learning phase (between [110,150] control steps) where the robot navigates in the environment basing only on the reflexive inherited behaviors that rely on contact sensors and short range target detector sensors, reaching the targets only if they are in its trajectory; (b) trajectory (between [628,749] control steps) when it has learnt to reach both the targets; (c)trajectories (between [1668,1825] control steps), when it has learnt to reach the yellow targets and to avoid the blue ones; an obstacle has been added in the environment.

of the synapses and improving the robot basic capabilities after a few trials.

Furthermore, the spiking processing performed by the network is described in Fig.6 in which the spikes emitted by the neurons that determine the behavior of the robot during the simulation are shown. Each control step corresponds to 333ms of simulation in which the network elaborates the sensory inputs. STDP allows to identify causal correlations between the spikes emitted by two connected neurons. In fact, it is possible to notice that, at the beginning of the learning phase, the robot is not able to correlate the spikes emitted by the V2N1Col1 neuron (yellow target detected on the right) to inter-neurons (i.e. VISLN2 and VISRN2 in Fig.6(a)), but only when a target is detected by the right low level sensor (TRN1Col1) the corresponding inter-neuron (VISLN2) is excited and therefore, since the inter-neuron are directly connected to the motor-neurons, the robot turns on the right. Whereas, after the learning phase, as it is shown in Fig.6(b) and in Fig.6(c), the robot has learnt that, when it sees a yellow target on the right and therefore the V2N1Col1 neuron emits spikes (Fig.6(b)), it has to turn on the right, in fact the VISRN2 neuron emits more spikes than the VISLN2 one. After this event, the robot sees the target in front of it (V1N1Col1 neuron emits spikes) and therefore a forward movement is executed until the target is reached (TLN1Col1 and TRN1Col1 neurons emit spikes). A similar case happens for the neurons related to the blue target (Fig.6(c)). Finally, at the end of the simulation, the robot has learnt to avoid blue targets. Fig.6(d), in fact, shows that, when a blue target is detected, for example, on the left (V1N1Col2 neuron emits spikes), the robot turns on the right (VISLN2 neuron emits more spikes than the VISRN2) and consequently the robot does not reach the target: in fact, the TLN1Col2 and TRN1Col2 neurons are not excited.

Multimedia materials on the experiment are available on line [4].

VI. Conclusions

In this paper a bio-inspired control algorithm for the navigation of a mobile robot, based on Spiking Neural Networks, have been discussed.

The purpose of the implemented algorithm is to allow the robot to navigate avoiding obstacles and reaching or avoiding specific objects that represent the targets for the robot, thanks to the interaction with the environment.

The proposed neural structure functionally inspired by the olfactory learning in the MBs in insect has been applied to control a bio-inspired hybrid robot, named TriBot.

The experimental results show the efficiency of the algorithm. The system is able to learn the association among visual features and basic behaviors through the STDP rule in a reasonable time. The approach presented in this work can provide efficient navigation control strategies with very simple unsupervised learning mechanisms. Further works will include new sensors and basic behaviors that will allow to extend the neural architecture.

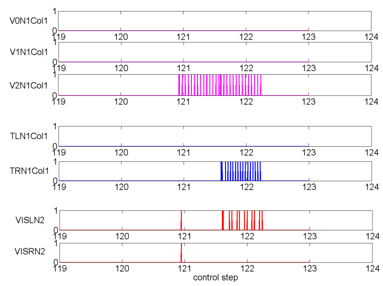
The learning structure is envisaged to be used within an insect brain computational model to solve more complex tasks in a real life scenario.

Acknowledgment

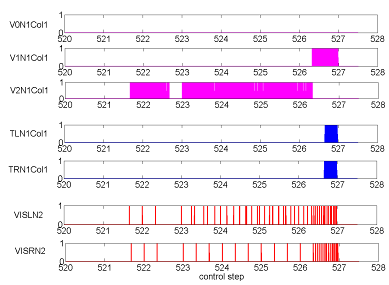
The authors acknowledge the support of the European Commission under the project SPARK II “Spatial-temporal patterns for action-oriented perception in roving robots II: an insect brain computational model”.

References

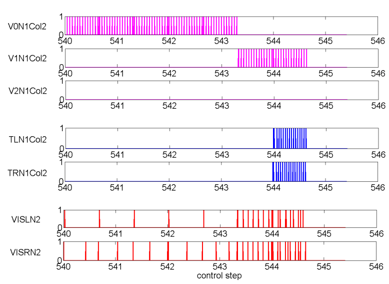
[1] M. A. Arbib, “The Handbook of Brain Theory and Neural Networks”, 1995.



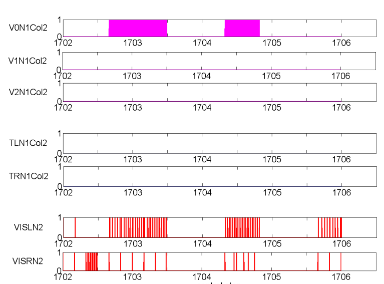
(a)



(b)



(c)



(d)

Fig. 6. Behavior of the network during the simulation. The spikes emitted by the neurons responsible for the robot behavior are reported. (a) Spikes emitted by the neurons related to the yellow colour during the learning phase (control step [119;124]). (b) Spikes emitted by the neurons related to the yellow colour (control step [520;528]) and (c) blue color (control step [540;546]) after the colour-target association has been reached. (d) Spikes emitted by the neurons related to the blue colour, when the robot has learnt to avoid the blue targets (control step [1702;1706]).

[2] J. Wessnitzer, B. Webb, “Multimodal sensory integration in insects - towards insect brain control architectures”, *Bioinsp. Biomim*, **1**, 63–75 (2006)

[3] P. Arena and L. Patan (Eds), “Spatial Temporal Patterns for Action Oriented Perception in Roving Robots”, *ISBN: 978-3-540-88463-7, Springer, Series: Cognitive Systems Monographs*, Vol. 1, 2009.

[4] TriBot Home Page, online at *http://www.spark2.diees.unict.it/ TribotI.html*.

[5] P. Arena, L. Fortuna, M. Frasca, L. Patané, “Learning anticipation via spiking networks: application to navigation control”, *IEEE Transaction on Neural Network*, Vol. 20, N. 2, pp. 202-216, February 2009.

[6] B. Gerber, H. Tanimoto, M. Heisenberg (2004), “An engram found? Evaluating the evidence from fruit flies”, *Curr Opin Neurobiol*, 14: 737744.

[7] Shepherd, G.M., “Neurobiology”, Oxford Univ. Press (1997).

[8] I. Pavlov, “Conditioned Reflexes: An Investigation of the Physiological Activity of the Cerebral Cortex”, Oxford University Press, London (1927)

[9] Bi G. and Poo M., “Synaptic Modifications in Cultured Hippocampal Neurons: Dependece on Spike Timing, Synaptic Strength and Postsynaptic Cell Type”, *The Journal of Neuroscience*, 18(24), 10464-10472, 1998.

[10] F. Alnajjar, K. Murase, “Self-Organization of Spiking Neural Network Generating Autonomous Behaviour in a Real Mobile Robot”, *International Conference on Computational Intelligence for Modelling, Control and Automation*, 2005.

[11] Hani Hagras, Anthony Pounds-Cornish, Martin Colley, Victor Callaghan, Graham Clarke, “Evolving Spiking Neural Network Controllers for Autonomous Robots”, *Proceedings of the 2004 IEEE International Conference on Robotics and Automation*, New Orleans, LA April 2004.

[12] N. Kubota, “A Spiking Neural Network for Behaviour Learning of A Mobile Robot in A Dynamic Environment”, *IEEE International Conference on Systems, Man and Cybernetics*, 2004.

[13] WHEGS Robot, online at *http://biorobots.cwru.edu/ projects/whegs/whegs.html*

[14] E. M. Izhikevich, “Which Model to Use for Cortical Spiking Neurons?”, *IEEE Transactions on Neural Networks*, Vol. 15, No. 5, 1063–1070, 2004.

[15] E. M. Izhikevich, “Simple Model of Spiking Neurons”, *IEEE Transactions on Neural Networks*, Vol. 14, No. 6, 1569–1572, 2003.

[16] U. R. Karmarkar,M. T. Najarian, and D. V. Buonomano, ”Mechanismus and significance of spike-timing dependent plasticity”, *Biol. Cybern.*, Vol. 87, pp. 373-382, 2002.

[17] S. Song, K. D. Miller, L. F. Abbott, “Competitive Hebbian learning through spike-timing-dependent plasticity”, *Nature Neurosci.*, Vol. 3, pp. 919–926, 2000.

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

[19] E.M. Izhikevich, “Solving the distal reward problem through linkage of STDP and dopamine signaling”, *Cerebral Cortex Advance*, 2007.

[20] G. Chechik, D. Horn, E. Ruppin, “Neuronal Regulation and Hebbian Learning”, *The handbook of brain theory and neural networks*, 2nd Edition, 2000.

[21] P.F.M.J. Verschure, B.J.A. Kröse, R. Pfeifer, “Distributed adaptive control: The self-organization of structured behaviour”, *Robotics and Autonomous Systems*, vol. 9, pp. 181-196, 1992.

[22] P.F.M.J. Verschure and R. Pfeifer, “Categorization, Representations, and the Dynamics of System-Environment Interaction: a case study in autonomous systems”, in J.A. Meyer, H. Roitblat, S. Wilson (Eds.) From Animals to Animats: *Proceedings of the Second International Conference on Simulation of Adaptive Behaviour*, Cambridge, MA, MIT Press, pp. 210-217, 1992.

[23] Datasheet ATMEL ATMEGA128 Microcontroller: *http://www.atmel.com/dyn/resources/prod_documents/doc2467.pdf*

[24] V. Braitenberg, “Vehicles: experiments in synthetic psycology”, MIT Press: Cambridge, MA, 1984.

[25] P. Arena, S. De Fiore, L. Patan, M. Pollino, C. Ventura, “STDP-based behaviour learning on the Tribot robot, *Proc. of Microtechnologies for the New Millennium (SPIE 09)*, Dresden (GERMANY), 4-6 May 2009, 7365-05.