Neural Networks 22 (2009) 801–811

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

2009 Special Issue

Cellular Nonlinear Networks for the emergence of perceptual states: Application to robot navigation control

Paolo Arena, Sebastiano De Fiore, Luca Patané *

Dipartimento di Ingegneria Elettrica, Elettronica e dei Sistemi, Universitá degli Studi di Catania, 95125 Catania, Italy

ARTICLE INFO

Article history:
Received 30 April 2009
Received in revised form 4 June 2009
Accepted 25 June 2009

Keywords:
Cognitive systems
Turing patterns
Reward-based learning
Roving robot

ABSTRACT

In this paper a new general purpose perceptual control architecture, based on nonlinear neural lattices, is presented and applied to solve robot navigation tasks. Insects show the ability to react to certain stimuli with simple reflexes, using direct sensory-motor pathways, which can be considered as basic behaviors, inherited and pre-wired. Relevant brain centres, known as Mushroom Bodies (MB) and Central Complex (CX) were recently identified in insects: though their functional details are not yet fully understood, it is known that they provide secondary pathways allowing the emergence of cognitive behaviors. These are gained through the coordination of the basic abilities to satisfy the insect's needs. Taking inspiration from this evidence, our architecture modulates, through a reinforcement learning, a set of competitive and concurrent basic behaviors in order to accomplish the task assigned through a reward function. The core of the architecture is constituted by the so-called Representation layer, used to create a concise picture of the current environment situation, fusing together different stimuli for the emergence of perceptual states. These perceptual states are steady state solutions of lattices of Reaction–Diffusion Cellular Nonlinear Networks (RD-CNN), designed to show Turing patterns. The exploitation of the dynamics of the multiple equilibria of the network is emphasized through the adaptive shaping of the basins of attraction for each emerged pattern. New experimental campaigns on standard robotic platforms are reported to demonstrate the potentiality and the effectiveness of the approach.

1. Introduction

Current approaches to the implementation of cognitive systems are mainly divided into two classes: the cognitivist approach based on symbolic information processing, and the emergent systems approach. The former often used Artificial Intelligence technique, while the latter focused on the exploitation of self-organization in dynamical systems. This is often based on bio-inspired solutions, relying on distributed networks mimicking the cerebral system. In some cases both approaches are used, creating hybrid architectures (Vernon, Metta, & Sandini, 2007).

Drawing inspiration from perceptual mechanisms of biological systems, machine perception researchers are starting to develop new perception schemes for roving robots. For example, Verschure and co-workers developed a perceptual scheme (Distributed Adaptive Control, DAC5) as a neural model for classical and operant conditioning (Verschure, Voegtlin, & Douglas, 2003). Recently Gnadt and Grossberg (2008) introduced SOVEREIGN, a neural architecture that can incrementally learn planned action sequences to carry out route-based navigation towards a rewarded goal. The architecture includes several interacting subsystems which model complementary cortical properties summarized in the *What* and *Where* processing streams. Other interesting approaches were proposed by Freeman and co-workers. They developed a dynamical model of the olfactory system, called K-sets (Freeman, 1987). A discrete implementation of Freemans K model (i.e. KA sets) was developed and applied to navigation control of autonomous agents (Harter & Kozma, 2005). The controller parameters have been learned through an evolutionary approach and also by using unsupervised learning strategies (Harter, 2005).

In this paper, following the paradigm known as Behavior-Based Robotics (Arkin, 1991), in which the perceptual process is considered tightly interconnected with the agent behavioral needs, perception has been treated as an emerging complex phenomenon. Here a large amount of heterogeneous information is fused to create an abstract and concise internal representation of the surrounding environment, which at the same time takes into account the needs and the motivation of the agent (Brooks, 1994), while the whole process is mediated through a behavioral-dependent internal state (Nolfi, 2002).

Starting from these considerations and taking into account the latest results in the field of neurobiology (Freeman, 2004) and the advancement in artificial cognitive system (Vernon et al., 2007),

* Corresponding author. Tel.: +39 0957382322.
E-mail addresses: parena@diees.unict.it (P. Arena), sdefiore@diees.unict.it (S. De Fiore), lpatane@diees.unict.it (L. Patané).

0893-6080/$ – see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.neunet.2009.06.024

we developed a general control architecture for implementing the sensing-perception-action cycle (Lynch, 1960) to be potentially applied to different robotic platforms involved in several missions in cluttered environments. To this aim, we borrowed from the insect world neural structures responsible for both simple and complex behaviors.

The internal representation of the external world, used for the action or behavior selection, is formalized by using Turing patterns (Murray, 2002; Turing, 1952). Classical examples of Turing patterns are animal coat patterns (stripes, spots and so on). In this work, Turing patterns are obtained in a nonlinear dynamical system, a Reaction–Diffusion CNN (RD-CNN) (Goras & Chua, 1995), as steady state conditions. More formally, they are attractors in complex nonlinear dynamical systems for particular sets of environmental stimuli and serve to modulate, through a reinforcement learning, competitive and concurrent basic behaviors. Learning is also introduced in the afferent layer to shape the basins of attraction of the Turing patterns in order to enhance this form of dynamic classification of the sensory events. This learning mechanism leads to the formation of abstract and flexible internal representations, mediated both by the environment and the agent needs. The second order cells within the RD-CNN mimic non-spiking neuron models, like neurons of group 12 in the pleural ganglia of the sea mollusk *Clione Limacina* (Orlovsky, Deliagina, & Grillner, 1999). These non-spiking neurons are enrolled when a sudden speed variation has to take place, induced by external or even, as argued in Arshavsky, Panchin, and Pavlova (1989), internal (e.g. humoral) motivations. These steady state plateau potentials in this neuron group lead to a suitable modulation of the animal motion, to fulfill a given motivation. From a structural point of view, in this work Turing patterns are generated within an array of non-spiking neurons in a RD-CNN. They are used to form percepts, i.e internal representations of the external world information. It should be pointed out that the same CNN cell neural structure, with a suitable modulation of its parameters, can generate spiking dynamics that were used to model the Central Pattern Generator in bio-inspired robots (Arena, Fortuna, & Branciforte, 1999). Therefore, the RD-CNN structure can be considered as the basic unit to generate the suitable neural, self-organizing dynamics at different levels in an artificial brain architecture. It should be noted that several VLSI analog implementations of RD-CNNs have been developed (Arena, Fortuna, Frasca, & Patané, 2005). Such chip prototypes are hosted within boards containing programmable digital hardware, in such a way that complex dynamics representing the solutions within the chip can be post processed allowing a real time implementation of the whole architecture for robot control.

In this work we assigned to the robot, as a simple case of study, a foraging task. To investigate the learning capability of the proposed architecture, both simulations in a virtual environment and experiments on a roving robot have been considered.

2. Control architecture

As in insects, the proposed perceptual architecture is organized in various control levels consisting of functional blocks, acting either at the same level, as competitors, or at distinct hierarchical levels showing the capability to learn more complex, experience-based behaviors (Wessnitzer & Webb, 2006).

The control architecture is reported in Fig. 1. It consists of series of parallel sensory-motor pathways (i.e. basic behaviors) that are triggered and controlled by specific sensory events in a reflexive way, giving the knowledge baseline to the system. Going up in the hierarchical scheme, two relevant centres of the insect brain are considered: the Mushroom Bodies (MB) and the Central Complex (CX). Both MB and CX are not yet well understood from a biological/neurogenetic point of view. However interesting

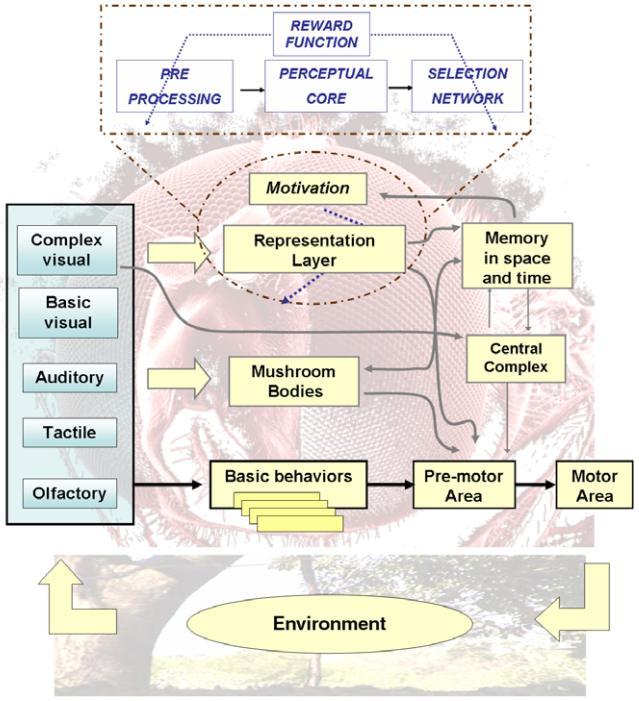

Fig. 1. Functional block diagram of the implemented control architecture. The interaction between the robot and the environment is realized by direct sensory-motor pathways, the *basic behaviors*, which are modulated by the representation layer. Mushroom Bodies (MB) and Central Complex (CX) are relevant centres of the insect brain devoted to temporal correlation, information storage and retrieval, and other functionality summarized in a correlation layer. Finally the high level functions of the representation layer consist of a *preprocessing block*, a *perceptual core*, a *selection network*, while the *Reward function* drives the learning process.

studies (Gronenberg & Lopez-Riquelme, 2004; Homberg, 1987; Wessnitzer & Webb, 2006) underlined how deeply these structures are involved in perceptual processes. In particular MBs are mainly devoted to the enhancement of causal relations arising among the basic behaviors, by exploiting the temporal correlation between sensory events; information storage and retrieval in the case of the olfaction sense; resolving contradictory cues through the visual sense by imposing continuation or adaptive termination of ongoing behavior. CX is instead responsible of integration of visual information, storing and retrieving information on objects and their position in space, controlling the step length in order to approach or avoid such objects; motor control, landmark orientation and navigation, orientation storage and others. Some of these functionalities have been already developed creating a correlation-based anticipation layer.

These aspects were treated separately in previous papers by using causal Hebbian rule in an array of spiking neurons for anticipation (Arena, Fortuna, Frasca, & Patané, 2009), and in Arena and Patané (2009) where memory structures based on Recurrent Neural Networks were considered. In this paper, for the sake of brevity, we briefly discuss only the high level representation layer, where perception is formed, considering that the anticipation layer can be added within this architecture to further enhance the capabilities.

As depicted in Fig. 1 the control process can be divided into functional blocks: at the lowest level, we place the parallel pathways representing the basic behaviors, each one triggered by a specific sensor; at a higher level we introduce a representation layer that processes all the sensory information in order to define the final behavior. At the highest layer we introduce a lattice of non-spiking neurons. This neural lattice shows distinct characteristics of complex dynamical systems. The emerging assemblies of neural states take on the meaning of percepts. These

ones are then associated to suitable modulations of the basic behaviors. This modulation is performed through an unsupervised learning process which creates associations among sensory stimuli and patterns. In this way, at the end of the leaning stage, each pattern represents a particular behavior modulation, while its trained basin of attraction represents the set of all the environment conditions, as recorded through the sensors, leading to the emergence of that particular behavior modulation. The modulation parameters associated with each pattern are learned through a reinforcement learning: here the reinforcement signal is provided by a motivation layer implementing the degree of satisfaction of the robot. This depends on the local satisfaction of the single basic behaviors with the addition of other terms that reflect the robot mission. The presence of additional information into the motivation layer, not used by the basic behaviors can be exploited by the Representation layer in order to increase the robot performance.

Memory is of course distributed in the whole architecture but a specific block has been also created (i.e. Memory in space and time in Fig. 1). This block develops a contextual layer, like in Verschure et al. (2003). Here sequences of successful emerged patterns are memorized to be retrieved when needed. Details about this memory structure are reported in Arena et al. (2007).

In such a way, as it happens in insects, the basic behaviors, which are often life-saving sensory-motor pathways, are progressively enriched with emergent capabilities which incrementally increase the animal skills. The main focus is therefore on the application of complex dynamics to obtain a proper, complex, context-learned modulation of the basic skills. This process is the main characteristic of our approach which makes it different from the other control strategies, based on the subsumption architecture proposed by Brooks (1986). The latter in fact, uses a high level approach to design both the basic behaviors and the coordination block. In our strategy, complex dynamical systems are successfully used. Both architectures use a behavioral decomposition of the system to exploit parallel computation, although the Subsumption network makes a rigid hierarchy among the basic behaviors: the lower ones cannot influence the upper ones, while the latter can act on the former. In our scheme, taking inspiration from the insect brain organization, all the basic behaviors are sensory-motor pathways elicited by only one sensory modality and on the same hierarchical level: knowledge is incrementally built upon their modulation, giving importance to one or the other, depending on the context. Under this perspective the proposed architecture resembles the Motor Schemas, introduced by Arkin (1991). Turing Patterns in RD-CNN are hosted, in our architecture, within a layer here called *Representation Layer*. The Representation Layer in our architecture does not refer to a place where a predictive model of the body–environment interaction is learned. This area is rather a layer where the single sensory-motor modalities, constituted by the parallel sensory-motor pathways, are modulated in a feedforward way, taking into account all the incoming sensory stimuli. This leads to the emergence of a contextually self-organizing activity, focusing at modulating the basic behaviors.

All the sets of environmentally driven multisensory information, leading to one rewarding behavior modulation, are collected into a unique basin of attraction. It is represented by its steady state condition, depicted as a pattern. This pattern is a binary image, suitable for a very compact coding. It should be noted that the number of different patterns that are able to emerge from the neural RD lattice could be very high (on the order of some hundreds in a square 4×4 network). So the number of different behavior modulations could be as large as needed to cope with very complicated and cluttered environments. The result of the behavior modulation leads to a particular robot motion, at each time t. This is formalized with a final action $A_F(t)$ that consists of a variable turning movement (rotation) and a fixed-length forward movement. The main characteristics of the cognitive architecture are described in the following subsections.

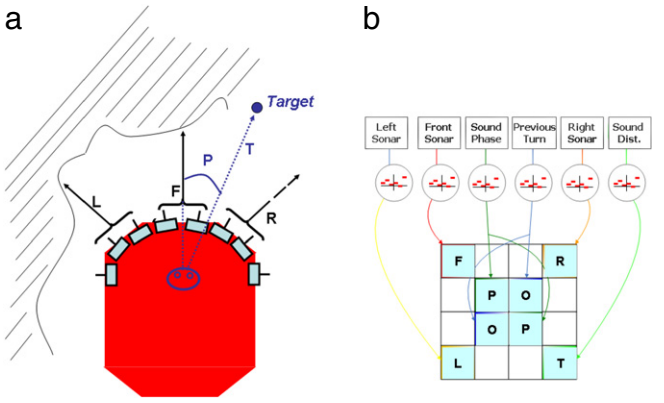

Fig. 2. (a) The robot acquires information from six sonar sensors grouped into three pairs (F: Front, L: Left, R: Right) and a target sensor providing the phase (P) and distance (T) between robot and the target. (b) Initialization for the first layer CNN cells in the representation layer. The corner cells are set by obstacle stimuli (Front, Left, Right obstacle distance sensors) and by the target distance sensor, if present. The central cells are set by the previous executed rotation (O) and by the angle between the robot heading and the robot–target direction.

2.1. Sensory block

To deal with the problem of autonomous navigation, the robot is provided with three distance sensors (covering the front, left and right hand sides of the robot) for obstacle detection. Moreover, the robot receives information on the angle between the robot orientation and the direction of the robot–target and, in some simulations, also on the distance between the robot and the target. A graphic overview of the sensory apparatus is sketched in Fig. 2.

2.2. Basic behaviors

With *basic behaviors*, we refer to some "genetically" prewired reflexes, triggered by specific sensory events through direct sensory-motor pathways. Referring to crickets, these behaviors are: the capability showed by crickets to recover heading during walking, called *optomotor reflex* (Böhm, Schildberger, & Huber, 1991); the female ability to follow the sound chirp emitted by a male, named *phonotaxis* (Webb & Scutt, 2000); and the ability to avoid obstacles, e.g. detected by the antennae.

At each time step t, the *optomotor reflex* tries to compensate for the previously executed rotation, as occurs in crickets that try to compensate leg asymmetry to maintain the heading during walk.

Even though a detailed neural network could be developed to carefully model the neural control system for such behavior (e.g. see Russo, Webb, Reeve, Arena, and Patané (2005)), in this work a very simple rule was adopted consisting of: $A_o(t) = -A_F(t-1)$, where $A_o(t)$ is the rotation triggered by the optomotor reflex at the time step t and $A_F(t-1)$ is the turn executed by the robot at the previous time step.

The *obstacle avoidance* behavior guides the robot in avoiding obstacles perceived using multiple distance sensors: $A_a(t) = f_a(d_F(t), d_L(t), d_R(t))$; here $A_a(t)$ is the rotation triggered by the obstacle avoidance, $f_a(\cdot)$ is a simplified version of the traditional potential field navigation algorithm (Borenstein & Koren, 1991) and $d_F(t)$, $d_L(t)$, $d_R(t)$ are the distances provided by the three distance sensors.

Finally, *phonotaxis* proposes a rotation $A_p(t)$, aiming to compensate for the phase between the robot heading and the robot–target direction: $A_p(t) = f_p(p(t))$, where $p(t)$ is the phase between the robot and the sound source. The function $f_p(\cdot)$, used in this application, is a simplified version of the model for phonotaxis behavior, reported in Horchler, Reeve, Webb, and Quinn (2004). Table 1 summaries the model variables used to describe the system basic behaviors.

Table 1
Summary of the model variables used to describe the basic behaviors and their meaning (see text for details).

$A_F(t)$	Final action
$A_o(t)$	Optomotor reflex
$A_p(t)$	Phonotaxis behavior
$A_a(t)$	Obstacle avoidance behavior
f_a	Potential field
f_p	Cricket inspired phonotaxis
$d_i(t)$	Distances robot–obstacle ($i = F, R, L$)
$p(t)$	Phase robot–target

2.3. Representation layer

The ability to interpret "situations" in terms of robot environment interaction (i.e. perception for action), is here considered as a *complex behavior*, growing up from the basic behaviors. The robot perceives using its sensory apparatus and processes at a cognitive level to optimize its behavior in relation to the mission assigned. The aim of the Representation layer, the highest control level within the whole cognitive process, is to achieve context dependent decisions. To this aim, all the available sensory modalities, each one separately being responsible of each single basic behavior, have to constitute the input to this layer. They are here incrementally transformed into environment *representations*, which lead to the modulation of the basic behaviors. These mechanisms are plastically modified by experience. In this work a CNN was designed to generate, on the basis of information coming from sensory events, Turing patterns as perceptual patterns. At the afferent (i.e. input) level, an unsupervised learning algorithm plastically shapes the basins of attraction of the Turing patterns in order to adjust the classification of the information with respect to the robot motivation.

The whole *representation layer* consists of a *preprocessing block*, a *perceptual core*, a *selection network* and a *motivation* layer, responsible for driving the learning process. Fig. 1 shows the main components of the representation layer.

2.3.1. Preprocessing block

The sensorial inputs, normalized in the range $[-1, 1]$, enter the preprocessing block: each stimulus is the input for a Sensing Neuron (*SN*) with piece-wise linear activation function, made-up, in this case, of 10 amplitude-varying steps learned in an unsupervised way as briefly explained in Section 2.3.4. Finally, each output of the *SNs* sets the initial condition for a cell of the nonlinear dynamical system that realizes the perceptual core of the Representation layer.

2.3.2. Perceptual core

The creation of a concise representation of the environment is crucial for the cognitive process, since it is the result of the dynamic processing of the external stimuli.

To implement this feature, we use a nonlinear partial differential equation, discretised in space as a neural lattice made-up of second order CNN cells, connected by local diffusion. This constitutes a two-layers RD-CNN, able to generate Turing patterns (Turing, 1952). The dimension of the network has been fixed to 4×4 on the basis of a previous work (Arena et al., 2007). Each cell $c(i, j)$ of the two-layers RD-CNN has state variables ($x_{1;i,j}$ for the first layer and $x_{2;i,j}$ for the second layer, with $i, j = 1, \ldots, 4$) and reads:

$$\begin{aligned}
\dot{x}_{1;i,j} &= -x_{1;i,j} + (1 + \mu + \varepsilon)y_{1;i,j} - sy_{2;i,j} + D_1\nabla^2 x_{1;i,j} \\
\dot{x}_{2;i,j} &= -x_{2;i,j} + sy_{1;i,j} + (1 + \mu - \varepsilon)y_{2;i,j} + D_2\nabla^2 x_{2;i,j} \\
y_{h;i,j} &= \frac{1}{2}(|x_{h;i,j} + 1| - |x_{h;i,j} - 1|) \qquad (1)
\end{aligned}$$

where $y_{h;i,j}$ ($h = 1, 2$) is the output of the layer h of the cell $c(i, j)$ and D_1, D_2, μ, ε and s are parameters of the model. To satisfy the analytical conditions to obtain Turing pattern the parameters have been set to: $\mu = -0.7$, $\varepsilon = 1.1$, $s = 0.9$, $D_1 = 0.05$, $D_2 = 15$, $\gamma = 1/D_1 = 20$ (Arena et al., 2007).

As shown in Fig. 2.b, the output of each *SN* sets the initial conditions for the state variable of two central cells or a corner cell, which have been proven to have higher control than the other cells (Arena et al., 2007). The initial conditions for the state variables of the second layer are set around zero for all the cells.

The RD-CNN evolves towards the condition in which all the state variables of the first layer, i.e. the $x_{1;i,j}$, saturate at a value greater than 1 or lesser than -1. In this case, each output variable $y_{1;i,j}$ will be either 1 or -1, a condition that we consider a Turing pattern.

To better understand how initial conditions influence the pattern emergence, we performed a simple experiment, where we set to zero the initial conditions for all the first layer cells except the top-right corner ($C(1; 4)$) and bottom-left ($C(4; 1)$) corner cells of a 4×4 lattice, whose initial conditions have been varied in $[-1; 1]$ range, mimicking two sensory inputs. The second layer cells are set to random values in the range [0:005; 0:005]. Fig. 3(a) shows the geometries of the basins of attraction for the 39 emerged patterns (represented by different colors), obtained by varying the initial conditions for the two cells above mentioned. At the end of the learning phase, executed with a simulation in which the two sensory inputs were associated to two distance-to-obstacle sensors, the number of basins of attraction and corresponding emerging patterns is tightly decreased to 15 (Fig. 3(b)). The effect of the unsupervised learning in the sensing block was to cluster the Turing patterns into a meaningful set of non-redundant internal states, and to adapt the internal states (shape of basins of attraction) to the robot motivation.

To simplify the successive processing, we associate a simple integer code for each Turing pattern as already discussed in Arena et al. (2007). The code is obtained converting the binary scheme of the pattern (i.e. sequence of black and white cells) into an integer value. This code is stored in a *Pattern Vector* at the first occurrence. Each element of the pattern vector contains the *Pattern Code* and the step of its last occurrence (*Occurrence Lag*). Once the external stimuli have been preprocessed, we reset the CNN, set again the initial conditions of the selected cells through the new outputs of the *SNs* (Fig. 2.b) and let the CNN re-evolve and generate a Turing pattern.

The effect in terms of trend of new emerged patterns during learning is shown in Fig. 4 where a typical result for a single run is reported. The use of Turing patterns as steady states of a dynamical system implies a form of sensor fusion, i.e. we synthesize heterogeneous sensory information into a single attractor. At each step, the information coming from sensors is fused to form a unique abstract and concise representation of the environment, as discussed in Section 2.

2.3.3. Selection network

The *Selection Network* associates each element q of the pattern vector with a set of three parameters (k_o^q, k_a^q, k_p^q). At the first occurrence of the pattern q, they are randomly chosen in the range [0, 1] with the constraint that: $k_o^q + k_a^q + k_p^q = 1$. Then, the parameters are modified under the effect of the learning process acting at the efferent (i.e. output) stage of the Representation layer as explained in the following. After completing the learning process and once the Turing pattern $q(t)$ has been generated at each time step t, the corresponding modulation parameters are selected and the behavior that emerges is the weighted sum of the actions suggested by the basic behaviors at that time: $A_F(t) = k_o^q \cdot A_o(t) + k_a^q \cdot A_a(t) + k_p^q \cdot A_p(t)$.

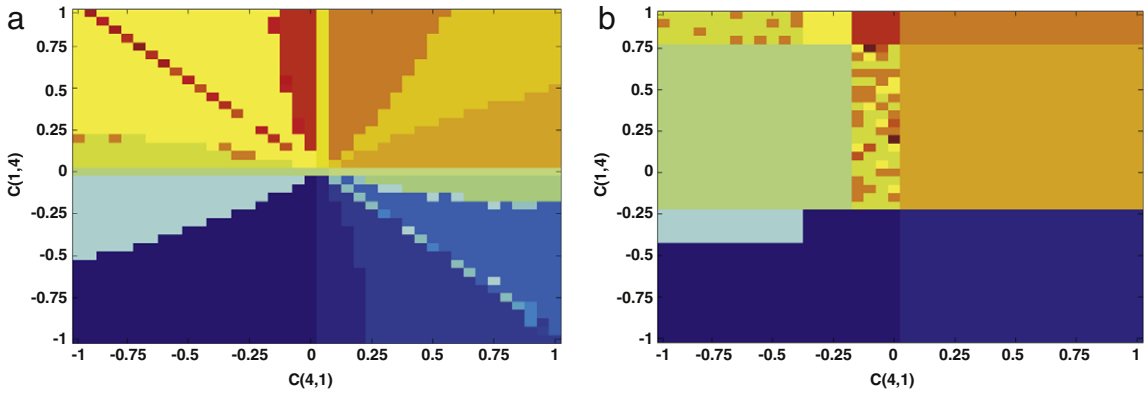

Fig. 3. (a) Initial Basins of attraction for the 39 patterns and (b) final shape of the basins of attraction for the 15 patterns obtained with the introduction of the learned sensing neurons. The patterns emerged by varying initial conditions for top-right corner cell (x-axis) and bottom-left corner cell (y-axis) of a 4×4 lattice in the range $[-1, 1]$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

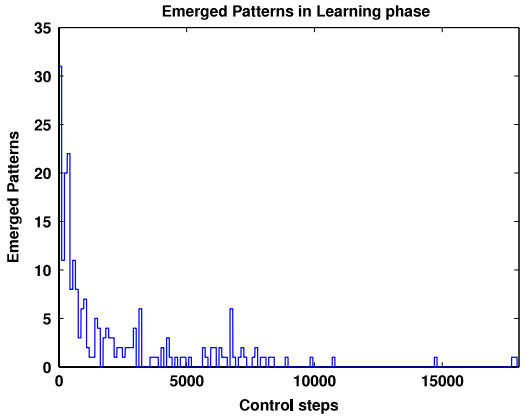

Fig. 4. Number of new patterns that emerge during learning when $\gamma = 20$. It is a typical result for a single run of the robot. The number is an average in windows of 100 steps.

2.3.4. Motivation layer and learning process

The association between Turing patterns and modulation parameters is learned through a reward-based reinforcement learning implemented by a simplified Motor Map (MM) (Arena et al., 2007; Schulten, 1992), whereas the fitness of each action is evaluated by means of a Reward Function (RF), defined as follows:

$$RF(t) = \sum_i h_i \cdot RF_i(t) \tag{2}$$

where RF_i represents the degree of satisfaction related to the basic behavior i where $i = o, a, p$, indicating optomotor, avoidance and phonotaxis reflex, respectively:

$$\begin{aligned} RF_o(t) &= r_o(|A_F(t-1)|) \\ RF_a(t) &= \sum_i r_i(e^{d_i(t)}) \\ RF_p(t) &= r_p(|p(t)|). \end{aligned} \tag{3}$$

Here $A_F(t)$ is the action performed at time t, $d_i(t)$ is the distance between the robot and the obstacle detected by the sensor i ($i = Front(F), Right(R), Left(L)$) and $p(t)$ is the phase between the robot orientation and robot–target direction. The goodness of the behavior can be evaluated at each step via the function $DRF(t) = RF(t) - RF(t-1)$. A positive (negative) value for $DRF(t)$ indicates a successful (unsuccessful) behavior. Successful behaviors are followed by reinforcement, like in Skinner's experiments (Skinner, 1974) in order to maximize the RF. In more details during learning, when the Turing pattern q emerges at the time step t, the behavior performed by the motor layer is:

$$A_F(t) = \sum_i (k_i^q + g_i^q(\xi)) \cdot A_i(t) \tag{4}$$

where $g_i^q(\xi)$ ($i = o, a, p$) are Gaussian variables (zero-mean and unitary variance), the variance (σ_q^2 associated with the pattern q) determines the range of the *random search* for the optimal modulation parameters. After the execution of the behavior defined in (4), the $DRF(t)$ is evaluated and, in case it is greater than the average increase in the RF generated by q, called b_q, the modulation parameters are updated in the direction suggested by the random variable according to:

$$k_i^q(new) = k_i^q(old) + \varepsilon g_i^q(\xi) \tag{5}$$

where $\varepsilon = 0.1$ is the learning rate. Furthermore, the variance of the Gaussian variable is decreased exponentially. In case $DRF < b_q$, the modulation parameters do not change.

If $DRF < 0$, the learning process acts on the afferent (input) association, realized by the SNs, between the stimuli and the initial conditions for the CNN cells aiming to establish the correct association between the sensory events and the internal representations (Turing patterns). In particular, our choice for the SNs activation function consists in an increasing function constituted by ten variable amplitude steps, θ_i ($1 \leq i \leq 10$), covering the whole input range $[-1, 1]$. At the beginning of the learning phase, all the steps have zero amplitude and, when we want to punish the system due to a $DRF < 0$, the step amplitudes are modified randomly in order to try and change the pattern. The idea is that, when the action associated with the previous situation is no longer able to make the robot succeed in accomplishing the current task, a new pattern should emerge and the suitable action to this new environmental condition has to be learned by the robot. In such a way the sensorial stimuli will be divided into classes, associating different situations with patterns that generate rewarding behaviors. In more detail, if the action associated with the currently emerged pattern is unsuccessful (i.e. $DRF(t) < 0$), then the learning algorithm for each SN acts as follows:

- determine which of the RF components has suffered the highest decrease (e.g. the component associated with the Front side obstacle detector);
- for the selected SN determine the step amplitude θ_i related to the current input value;
- extract a number rnd from a zero-mean, uniformly distributed random variable r;

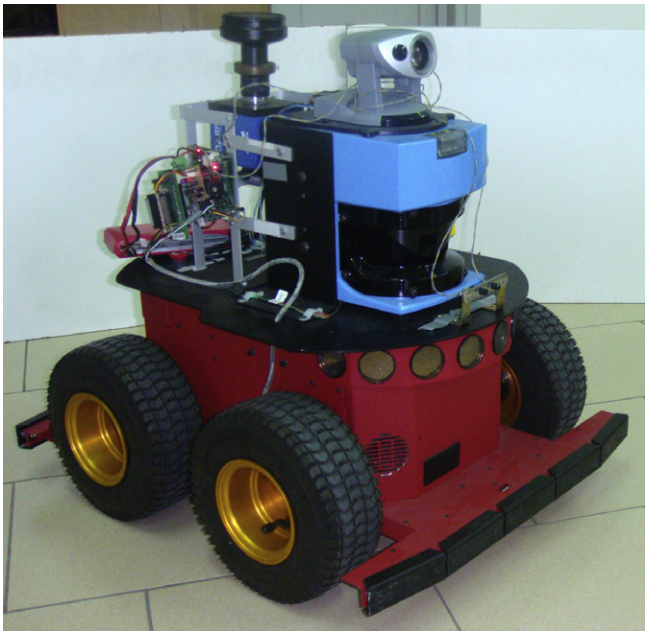

Fig. 5. The roving robot P3AT.

- the step amplitude θ_j is modified as: $\theta_i(new) = \theta_i(old) + rnd$, provided that it lies in the range $[-3, 3]$, compatible with the state variable dynamics for the CNN cells.

To guarantee the convergence of the algorithm, the variable *rnd* varies in the range $[-m, m]$ where m, initially sets to 0.5, decreases at each step with an aging coefficient $m(new) = 0.999 \cdot m(old)$. The result is that the association between sensorial stimuli and Turing patterns is dynamically tuned by modulating the basins of attraction of the steady state patterns. The effect is that, at the beginning of the learning phase, a lot of pattern-action associations arise which are stabilized at later stages. This strategy, already effective, can be improved by including the dependence on the Reward function fluctuations. More details on the whole mathematical model are given in Arena et al. (2007), whereas an initial assessment of the application to a roving platform can be found in Arena, Fortuna, Lombardo, and Patané (2008).

3. The roving robot P3-AT

The robot used for the experimental set-up is a standard platform, the Pioneer P3-AT robot built by *MobileRobots inc*. It is a classic four wheeled rover controlled through a differential drive system, using encoders with inertial correction to compensate for skid steering. The robot is equipped with an embedded computer, wireless Ethernet-based communication, a laser scanner, a compass sensor and a pan-tilt actuated color camera. Moreover it is equipped with eight forward sonars that sense obstacles from 15 cm to 5 m and five bumpers for collision detection.

The robot can be controlled through a suite of library (i.e. the ARIA library) and a 2D virtual simulation environment, named *MobileSim*, can be used instead of the real robot in a transparent way using the same control library.

As shown is Fig. 5, we have customized the standard configuration including a hearing circuit, a CNN-based camera with panoramic lens and a gray-scale color sensor placed on the bottom of the robot, used as a low level target sensor, to detect black spots on the ground. These additional sensors are managed from the onboard computer using a microcontroller-based bridge.

In the experiments reported in the following section, part of the robot sensory system was used to develop the three basic behaviors taken into consideration (see Section 2.2). In particular for the obstacle avoidance behavior both contact sensors (i.e. front and rear bumpers) and distance sensors (i.e. ring of sonars) were considered. To include the optomotor reflex, the gyroscope embedded on the robot was used. Finally, the phonotaxis behavior was reproduced using an *ad hoc* built cricket-calling-song producing circuit. A gray-scale color sensor was also used to detect the successful arrival at a sound target.

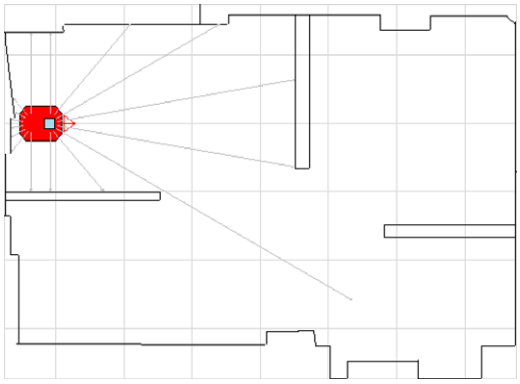

Fig. 6. Environment used for the robot simulations. The lines departing from the robot simulate the ring of sonar.

The hearing sensor is the most interesting part of the sensory apparatus: it allows efficient localization of a specific sound source with a very simple analog circuit. It is inspired by phonotaxis in crickets: female crickets are able to recognize the species specific pattern of male calling song, produced by opening and closing their wings, and move towards it. For *Gryllus bimaculatus*, these songs consist of four 20 ms syllables of 4.7 kHz sound waves, separated by 20 ms intervals, which make up a "chirp", produced several times a second. Females appear to be particularly selective for the repetition rate of syllables within each chirp. The hearing circuit (Webb & Scutt, 2000) consists of two microphones and a circuit board fine-tuned to the carrier frequency of the cricket song. The output from each ear is an analog signal in the range from 0 to 5 volts. The input to the circuit is given by two microphones separated by a distance equivalent to a quarter of the wavelength of the carrier frequency (i.e. 18 mm).

In the simulation environment (i.e. MobileSim), the whole sensory system was modeled taking into consideration specific characteristics of each sensor: detection range, time response, resolution and others.

4. Simulation results

4.1. Simulation set-up

The software simulation environment, developed in $C++$, allows us to interface with the P3-AT robot and its simulation environment (MobileSim). The arena used consists of three rooms as depicted in Fig. 6, with six targets randomly placed. The simulated environment reproduces a real environment with dimensions of about 7.5×4.5 m. The map was acquired using the scanner laser equipped on the real robot. The simulated robot is equipped with eight sonar sensors but only six of them were used, covering a range of $[-50°, 50°]$ with respect to the direction of motion. Moreover, the six sonar are considered as three pairs (left, front and right) and the minimum distance value acquired for each pair (i.e. nearest object for each side) is processed by the control architecture. The target sensor provides the distance from the active target and the phase between the robot orientation and the robot–target direction. The target sensor simulates the hearing board equipped on the real robot. It should be noted that, for all the distance sensors, the output is saturated to the limit of the

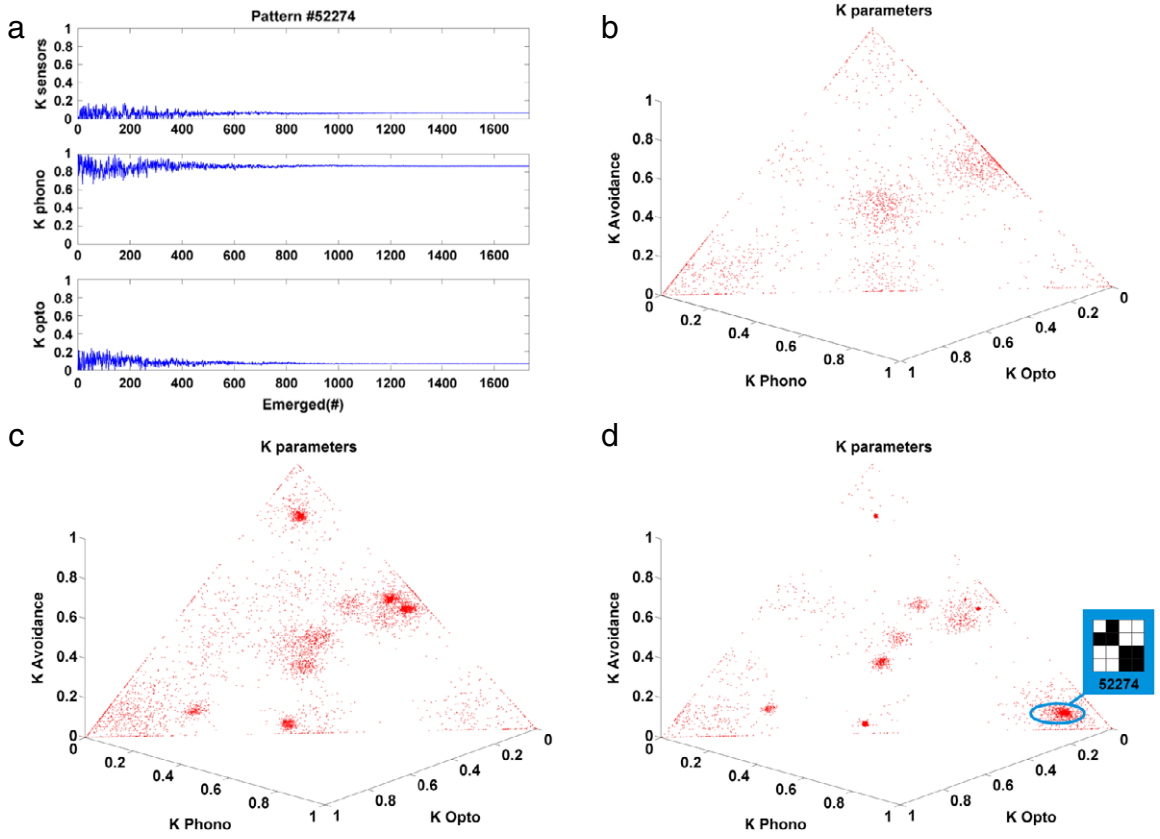

Fig. 7. (a) Evolution of the $k_i^f q_1$ for the emerged pattern ($q_1 = 52\,274$). The evolution of the modulation parameters is shown in (b–d) where the solutions adopted at the beginning of the learning stage (b), between 5000 and 10 000 movements (c) and for the last 5000 movements are shown. In the last picture it is also indicated the region associated with the pattern 52 274.

detection range, so even if no obstacles are detected, the output of the sensor would be 5 m. The target sensor has a range of 3 m and a visual conus of $[-90°, 90°]$. All the sensor outputs are scaled in the range $[-1, 1]$. The components of the *RF* in Eq. (3) were heuristically defined as: $r_o(t) = -A_F(t-1)$, $r_p(t) = -|p(t)|$, $r_F(t) = -e^{-8(d_F(t)+1)}$, $r_L(t) = -e^{-8(d_L(t)+1)}$, $r_R(t) = -e^{-8(d_R(t)+1)}$, where $d_F(t)$, $d_R(t)$, $d_L(t)$ are the distances detected by the sensors F, R, L, while $p(t)$ is the angle between the robot heading and the robot–target direction and $A_F(t-1)$ is the rotation made by the robot in the time step $t-1$. In the following simulations, the choice for the other parameters in Eq. (2) is $h_o = 1$, $h_a = 10$, $h_p = 10$. In this way more importance is given to the contribution of the obstacle information than to the target one, because the former is crucial to preserve the robot integrity. In particular the output coming from the front side obstacle sensor has the greatest weight in the *RF*. Through the definition of this reward function, we give to the robot knowledge about the task to be fulfilled, but it has no *a priori* knowledge about the correct way to interact with the environment. So the phase of the actions associated with each pattern is randomly initialized within the range $[-20°, 20°]$. The forward movement performed for each action has a fixed length of 25 cm.

4.2. Learning phase

As far as the simulated robot is concerned, the task assigned to the robot consists of aiming for a target and avoiding obstacles along the way. When the target is found it is switched off and another target appears in the arena. The learning phase lasts until one of the two following conditions occurs: either the a_q averaged on the last 1000 patterns drops below 10^{-4}, or after 5000 targets have been found. At the beginning of the learning phase, the robot randomly modulates the basic behaviors, due to the random initialization of the modulation parameters k_i^q ($i = a, o, p$), which determine the robot heading. During the learning process, the Motor Map-like algorithm corrects the parameters associated with each pattern. Fig. 7(a) shows the evolution of the $k_i^{q_1}$ for the pattern, i.e. $q_1 = 52274$. The modulation parameters associated to all the emerged patterns used in the first part, during and at the end of the learning phase (for a total number of 15 000 actions divided in three blocks of 5000) is shown in Fig. 7(b)–(d).

This evolution is typical in several other simulations (for other results see Arena, De Fiore, Lombardo, and Patané (2009)), taken into account and gives an idea of the clustering process that occurs during learning, where the basins of attraction of the Turing patterns that are associated to the modulation parameters, change in space and time leading to the emergence of specific rewarding behavior combinations. For example it is interesting to notice that the pattern q_1 in this simulation is associated to a behavior mainly guided by phonotaxis.

4.3. Testing phase

To evaluate the improvement of performance obtained during the learning process, we compared the result of the learned structure with other solutions: constant modulation parameters and randomly chosen modulation parameters. The constant behavior modulation parameters were chosen through a manual tuning aimed at optimizing the global performance of the robot. The parameters used in the following experiments are: $K_a = 0.35$, $K_p = 0.2$, $K_o = 0.05$. The randomly chosen modulation parameters gives an idea of the behavior of the robot at the beginning of the learning phase when the behavior modulation is initialized with random values.

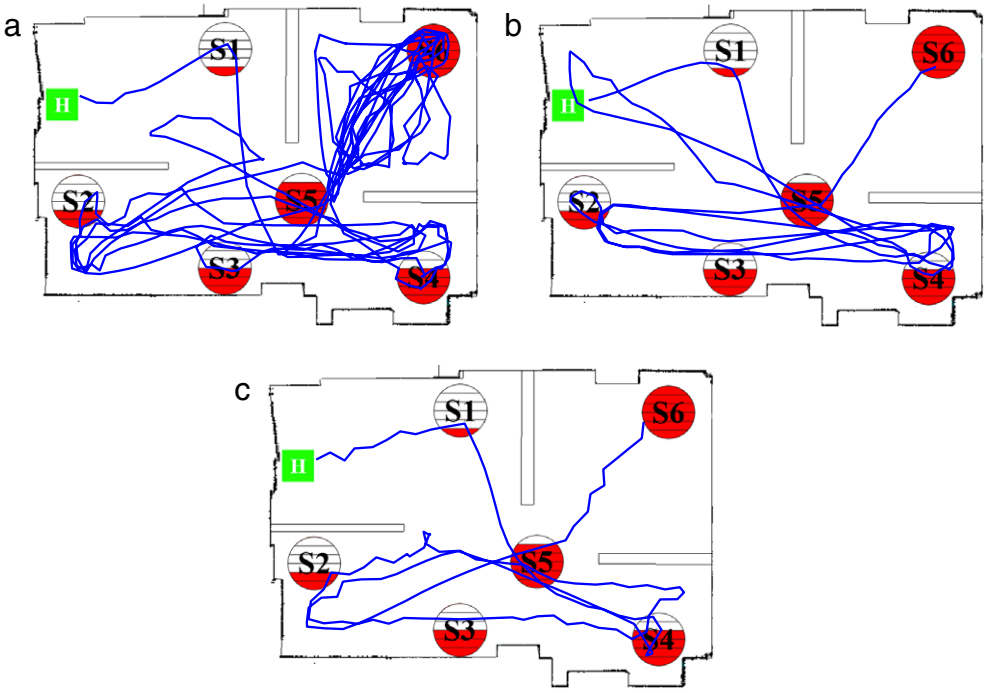

Fig. 8. Trajectories followed by the robot controlled through: random (a), fixed (b) and learned (c) parameters for the sequence P0. H is the starting point and the sequence is: P0(1, 2, 3, 4, 5, 6).

Table 2
Simulation results, number of actions needed to retrieve all the six targets in a given order and improvement with respect to the worst case (i.e. Random modulation). The sequences are: P0(1, 2, 3, 4, 5, 6); P1(2, 4, 6, 1, 3, 5); P2(1, 3, 5, 2, 4, 6); P3(3, 5, 2, 4, 6, 1).

Sequence	Number of Actions			Improvement (%)		
	Random	Fixed	Learned	Random	Fixed	Learned
P0	587	256	146	0	56.4	75.1
P1	455	303	299	0	33.4	34.3
P2	297	287	231	0	3.3	22.2
P3	574	524	418	0	8.7	27.2

Table 3
Simulation results, number of collisions that occurs during the target retrieving process and improvement with respect to the worst case (i.e. Random or Fixed modulation).

Sequence	Number of collisions			Improvement (%)		
	Random	Fixed	Learned	Random	Fixed	Learned
P0	55	35	20	0	36.4	63.6
P1	93	65	32	0	30.1	65.6
P2	44	46	26	4.3	0	43.5
P3	89	120	58	25.8	0	51.7

For the performance validation four different sequences of activation for the six targets have been used. The compared results, in terms of number of steps to complete the sequence and number of collisions, are reported in Tables 2 and 3. The trajectories followed during the experiments for the target activation sequence P0 are reported in Fig. 8.

The learning process leads to a significant reduction of actions needed to complete a sequence of targets searching, demonstrating the effectiveness of the control architecture and its capability to generalize the representations. In particular, this feature has been proven by performing the test in a scenario that is different from that one used during learning. Fig. 9 shows examples of trajectories followed during the testing phase in case of fixed, random and learned modulation parameters.

To further analyze the performance of the control architecture, a statistical analysis was performed in the environment shown in Fig. 9. A series of ten simulations for each control strategy was carried out modifying the initial orientation of the robot in the environment. The results for the fixed and learned modulation parameters are reported in Table 4. Also in these experiments, the performance improvement is evident in terms of path length to find the sequence of targets and robustness of the behavior, which can be observed by the low value of the standard deviation in the path length for the different trials. It should be noted that using the fixed modulation parameters as a benchmark, means to eliminate the RD-CNN from the architecture, and exploiting only the basic behaviors, whose weight in the action selection is constant throughout the experiment. The importance of the representation layer, introduced in this work, is to add to the architecture incremental capabilities of building knowledge, based on the environment. At the beginning of the learning phase, the role of the Turing Pattern Generator is negligible, and the robot moves only according to the basic behaviors. As learning proceeds, the robot acquires the capability to exploit the space-varying combination of the basic behavior to improve its performance in relation to its motivation.

Table 4
Comparison between fixed and learned modulation parameters in reaching the sequence of targets shown in Fig. 9. Both the mean value and the standard deviation for path length to reach the targets, and number of collisions are better when the Representation layer is active.

	Fixed		Learned	
	Mean value	Standard deviation	Mean value	Standard deviation
Number of steps	170	9	62.3	3
Number of collisions	15.7	1.5	2.3	1

The same architecture can deal with a moving target as shown in Fig. 10. The robot and the target start from two different positions, and while the former has a constant speed of 25 cm/action, the latter change its speed in the range 18–40 cm/action. The trajectories followed are shown in Fig. 10(a) while the robot–target distance during the simulation is shown in Fig. 10(b).

To better understand the sequence of patterns (i.e. combination of behaviors) used during a target retrieving, an example is shown

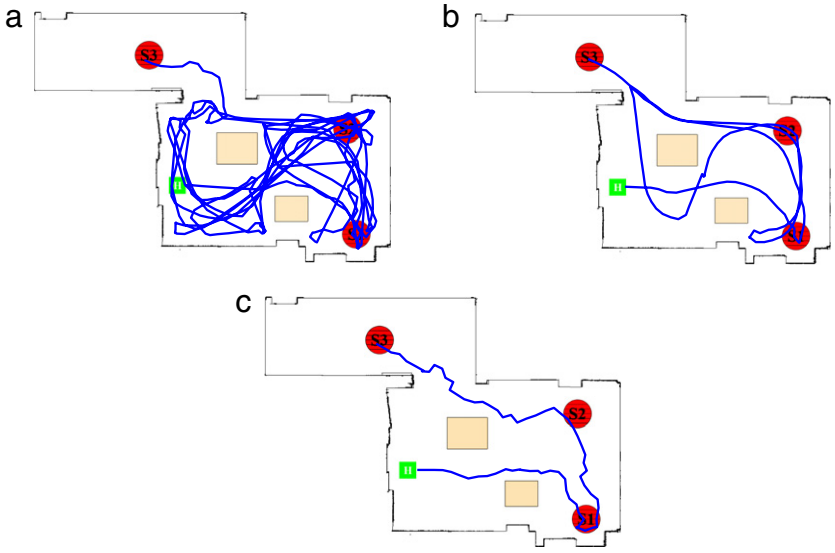

Fig. 9. Trajectories in a different arena in the case of constant (a), randomly chosen (b) and learned (c) modulation parameters. The first two strategies (a–b) take a lot of time to reach the target and suffer from multiple collisions. From the learned modulation parameters, a very straightforward, although safe, behavior emerges.

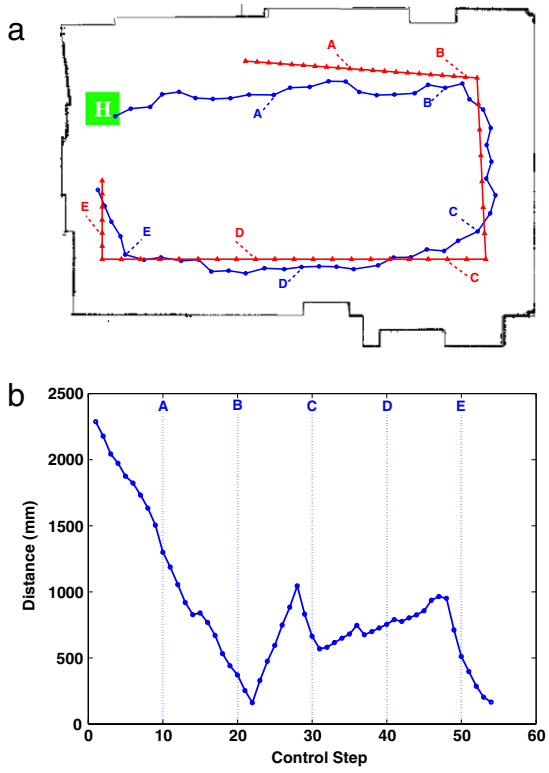

Fig. 10. Moving target simulation. (a) Trajectory followed by the robot (circle marked line) following a moving target (triangle marked line). (b) Trend of the robot–target distance during the simulation.

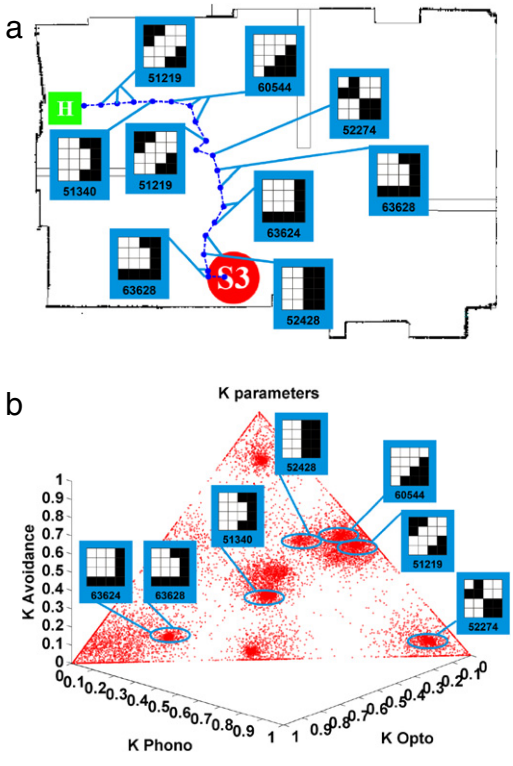

Fig. 11. (a) Sequence of patterns used during a target reaching task. (b) Each pattern corresponds to a behavior modulation.

in Fig. 11. At the beginning of the trajectory the robot persists in the chosen behavior combination until the environment conditions lead to a *jump* into the basin of attraction of another pattern, more adapt to represent the new situation. In this simulation about ten different behavior modulations are used to fulfill the assigned goal.

5. Experimental results

To cross-check experimentally the promising results obtained in numerical simulations we use the roving robot P3-AT moving

Fig. 12. Experimental set-up. The rover P3-AT facing an environment with two targets.

Table 5
Experimental results, number of actions needed to retrieve three targets in a given order and improvement with respect to the worst case (i.e. Random modulation).

Sequence	Path length (m)			Improvement (%)		
	Random	Fixed	Learned	Random	Fixed	Learned
P	64	24	17.5	0	62.5	72.7

Table 6
Experimental results, number of collisions that occurs during the target retrieving process and improvement with respect to the worst case (i.e. Random modulation).

Sequence	Number of collisions			Improvement (%)		
	Random	Fixed	Learned	Random	Fixed	Learned
P	11	6	2	0	45.4	81.8

in a real environment. Fig. 12 shows the experimental set-up: the robot is placed in an environment with two targets. The trajectories followed by the robot are reported in Fig. 13. For the complete architecture the same behavior modulation parameters obtained through the learning process in simulation were used for the tests with the real robot. The obtained performance are in line with the simulation results as reported in Table 5 for the path length and in Table 6 for the number of collisions.

It should be noted that when the robot is looking for the second target, after the first one is retrieved and switched off, the phono sensory system does not allow the detection of $S2$ because it is out of the sensor detection range (i.e. $[-90°, +90°]$). This is the reason for the long path followed to reach the second target also with the adaptive behavior modulation.

Videos, including simulations and real robot experiments are available on the web (Arena & Patané, 2008).

6. Remarks and conclusions

In this paper a new control architecture for the sensing-perception-action loop in robots is described and validated through simulations and experiments for autonomous navigation on a standard robotic platform. The control architecture is based on some predefined basic abilities, called basic behaviors, which are modulated by the *Representation Layer*. The Representation Layer learns to associate sets of sensory events with specific Turing patterns, and tunes modulation parameters through reinforcement learning to perform goal-directed behaviors. Here, unlike similar approaches referring to behavior-based robotics, we used complex dynamics to explore attractor based nonlinear computation (another relevant approach to navigation control can be found in Arena, Fortuna, De Fiore, and Patané (2008)) and a simple reward-based learning, to associate rewarding behavior modulation to contextual information coming from sensors. The whole sensory system depicts the environment scene as perceived by the robot. It is clear that within this information, the salient details about the robot body and position in the environment are naturally used to achieve an efficient, embodied and situated knowledge. It should be underlined that algorithms dedicated to handling navigation tasks could even give better results: the potentiality of our approach lies in its generality. In fact the approach can be easily migrated to other robotic platforms, redefining the basic behaviors, and to other applications, redesigning the reward function. The approach, for example, is being actually applied to a more complex structure, an hexapod robot (Arena & Patané, 2005), where the control actions are much more complex, and the basic behaviors could include, for instance, not only avoiding obstacles by turning, but also climbing over steps. In this case patterns can indicate the particular scheme of leg motions, which should be applied in front of particular environment conditions. Presently the use of complex dynamics to achieve contextualization does not enable the capability to make prediction on sequences of behaviors useful to reach the target. Indeed this could be inserted very easily by implementing chains of successful behavior modulations, but we are currently working at exploiting the complex dynamics within the Turing Pattern generator to include prediction capabilities. The above described framework is suitable to be included in a more complex bio-inspired architecture aiming to emulate an insect brain at least from a functional point of view. A wider set of heterogeneous sensors such as cameras could be included.

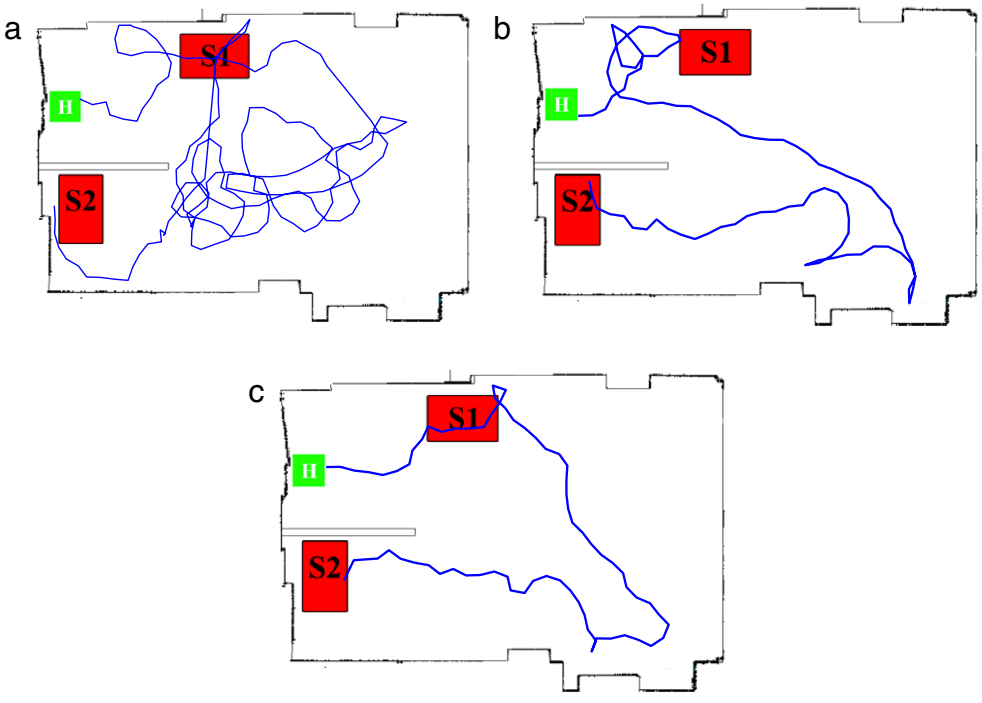

Fig. 13. Trajectories obtained with the real robot in the case of constant (a), randomly chosen (b) and learned (c) modulation parameters.

Acknowledgements

The authors acknowledge the support of the EU Commission under the project SPARK II and the precious collaboration with Professor R. Strauss, Institute of Zoology, Johannes Gutenberg Univ. of Mainz.

References

Arena, P., Crucitti, P., Fortuna, L., Frasca, M., Lombardo, D., & Patané, L. (2007). Turing patterns in RD-CNNs for the emergence of perceptual states in roving robots. *International Journal of Bifurcation and Chaos*, *17*(1), 107–127.

Arena, P., De Fiore, S., Lombardo, D., & Patané, L. (2009). Emergence of perceptual states in nonlinear lattices: a new computational model for perception. In *Proc. of int. joint conference on neural networks*.

Arena, P., Fortuna, L., & Branciforte, M. (1999). Reaction–diffusion CNN algorithms to generate and control artificial locomotion. *IEEE Transactions on Circuits and Systems, I*, *46*(2), 259–266.

Arena, P., Fortuna, L., De Fiore, S., & Patané, L. (2008). Perception-action map learning in controlled multiscroll system applied to robot navigation. *Chaos*, *18*, 1–16.

Arena, P., Fortuna, L., Frasca, M., & Patané, L. (2005). A CNN-based chip for robot locomotion control. *IEEE Transactions on Circuits and Systems, I*, *52*(9), 1862–1871.

Arena, P., Fortuna, L., Frasca, M., & Patané, L. (2009). Learning anticipation via spiking networks: Application to navigation control. *IEEE Transactions on Neural Networks*, *20*(2), 202–216.

Arena, P., Fortuna, L., Lombardo, D., & Patané, L. (2008). Perception for action: Dynamic spatiotemporal patterns applied on a roving robot. *Adaptive Behavior*, *16*(2–3), 104–121.

Arena, P., & Patané, L. (2005). *Robot gallery*. Retrieved May 8, 2009, from EU Project SPARK I Web site: www.spark.diees.unict.it/Robot.html.

Arena, P., & Patané, L. (2008). *Robot gallery*. Retrieved May 8, 2009, from EU Project SPARK II, Web site: www.spark2.diees.unict.it/Robot.html.

Arena, P., & Patané, L. (Eds.). (2009). *Cognitive systems monographs*: *Vol. 1. Spatial temporal patterns for action oriented perception in roving robots*. Springer, ISBN: 978-3-540-88463-7.

Arkin, R. C. (1991). *Behaviour based robotics*. Cambridge: MIT Press.

Arshavsky, Yu. I, Orlovsky, G. N., Panchin, Yu. V., & Pavlova, G. A. (1989). Control of locomotion in marine mollusc Clione limacina VII. Reexamination of type 12 interneurons. *Experimental Brain Research*, *78*, 398–406.

Böhm, H., Schildberger, K., & Huber, F. (1991). Visual and acoustic course control in the cricket *Gryllus-bimaculatus*. *Journal of Experimental Biology*, *159*, 235–248.

Borenstein, J., & Koren, Y. (1991). The vector field histogram fast obstacle avoidance for mobile robots. *IEEE Journal of Robotics and Automation*, *7*, 278–288.

Brooks, R. A. (1986). A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, *2*(1), 14–23.

Brooks, R.A. (1994). Intelligence without reason. In J. Mylopoulos, R. Reiter (Eds.), *Proc. of 12th int. conference on artificial intelligence*.

Freeman, W. J. (1987). Simulation of chaotic EEG patterns with a dynamic model of the olfactory system. *Biological Cybernatics*, *56*, 139–150.

Freeman, W. J. (2004). How and why brains create meaning from sensory information. *International Journal of Bifurcation and Chaos*, *14*, 515–530.

Gnadt, W., & Grossberg, S. (2008). SOVEREIGN: An autonomous neural system for incrementally learning planned action sequences to navigate towards a rewarded goal. *Neural Networks*, *21*, 699–758.

Goras, L., & Chua, L. (1995). Turing Patterns in CNNs - Part I, II. *IEEE Transactions on Circuits and Systems, I*, *42*, 602–626.

Gronenberg, W., & Lopez-Riquelme, G. O. (2004). Multisensory convergence in the mushroom bodies of ants and bees. *Acta Biologica Hungarica*, *55*, 31–37.

Harter, D. (2005). Evolving neurodynamics controllers for autonomous robots. *International Joint Conference on Neural Networks*, 137–142.

Harter, D., & Kozma, R. (2005). Chaotic Neurodynamics for autonomous agents. *IEEE Transactions on Neural Networks*, *16*(3), 565–579.

Homberg, U. (1987). Structure and functions of the central complex in insects. In AP Gupta (Ed.)., *Arthropod brain, its evolution, development, structure and functions* (pp. 347–367). New York: Wiley.

Horchler, A., Reeve, R., Webb, B., & Quinn, R. (2004). Robot phonotaxis in the wild: A biologically inspired approach to outdoor sound localization. *Advanced Robotics*, *18*(8), 801–816.

Lynch, K. (1960). *The image of the city*. Cambridge: MIT Press.

Murray, J. D. (2002). *Mathematical biology I: An introduction* (3rd ed.). NJ: Springer-Verlag.

Nolfi, S. (2002). Power and limits of reactive agents. *Neurocomputing*, *42*(1), 119–145.

Orlovsky, G., Deliagina, T. G., & Grillner, S. (1999). Neuronal control of locomotion: From mollusc to man. *Oxford Neuroscience*.

Russo, P., Webb, B., Reeve, R., Arena, P., & Patané, L. (2005). A cricket-inspired neural network for feedforward compensation and multisensory integration. In *IEEE conference on decision and control and European control conference 2005*.

Schulten, K. (1992). Theoretical biophysics of living systems. In H. Ritter, T. Martinetz, & K. Schulten (Eds.), *Neural computation and self-organizing maps: An introduction*. New York: Addison-Wesley.

Skinner, B. F. (1974). *About behaviorism*. New York: Alfred Knopf.

Turing, A. M. (1952). The chemical basis of morphogenesis. *Philosophical Transactions of Royal Society, London B*, *237*, 37–72.

Vernon, D., Metta, G., & Sandini, G. (2007). A survey of artificial cognitive system: Implications for the autonomous development of mental capabilities in computational agents. *IEEE Transactions on Evolutionary Computation*, *11*(2), 151–180.

Verschure, P. F. M. J., Voegtlin, T., & Douglas, R. J. (2003). Environmentally mediated synergy between perception and behaviour in mobile robots. *Nature*, *425*, 620–624.

Webb, B., & Scutt, T. (2000). A simple latency dependent spiking neuron model of cricket phonotaxis. *Biological Cybernetics*, *82*(3), 247–269.

Wessnitzer, J., & Webb, B. (2006). Multimodal sensory integration in insects — towards insect brain control architectures. *Bioinspiration and Biomimetics*, *1*, 63–75.