A new bio-inspired perceptual control architecture applied to solve navigation tasks

P. Arena[a], S. De Fiore[a], L. Patané[a] and A. Vitanza[a].

[a] Dipartimento di Ingegneria Elettrica, Elettronica e dei Sistemi, Universitá degli Studi di Catania, Viale A. Doria 6, 95125 Catania, Italy.

ABSTRACT

In this paper a new general purpose perceptual control architecture is presented and applied to robot navigation in cluttered environments. In nature, insects show the ability to react to certain stimuli with simple reflexes using direct sensory-motor pathways, which can be considered as basic behaviors, while high brain regions provide secondary pathway allowing the emergence of a cognitive behavior which modulates the basic abilities. 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 Representation layer, where different stimuli, triggering competitive reflexes, are fused to form a unique abstract picture of the environment. The representation is formalized by means of Reaction-Diffusion nonlinear partial differential equations, under the paradigm of the Cellular Neural Networks, whose dynamics converges to steady-state Turing patterns. A suitable unsupervised learning, introduced at the afferent (input) stage, leads to the shaping of the basins of attractions of the Turing patterns in order to incrementally drive the association between sensor stimuli and patterns. In this way, at the end of the leaning stage, each pattern is characteristic of a particular behavior modulation, while its trained basin of attraction contains the set of all the environment conditions, as recorded through the sensors, leading to the emergence of that particular behavior modulation. Robot simulations are reported to demonstrate the potentiality and the effectiveness of the approach.

1. INTRODUCTION

It is well know as Robotics is a multidisciplinary research field that nowadays is strongly related to perceptual and cognitive systems. In this paper a new action oriented perceptual architecture is proposed and applied to solve navigation tasks in cluttered environment.

Perception is here considered as an emerging complex phenomenon, where 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,[1] while the whole process is mediated through a behavioral-dependent internal state.[2]

Bio-inspired solutions are continuously proposed and applied to robotics to solve complex tasks exploring simple strategies used by living system. In particular insects are a wonderful example of simplicity mixed to richness in term of behaviours shown. Moreover insects show the ability to react to certain stimuli with simple reflexes using direct sensory-motor pathways, which can be considered as basic behaviors, while high brain regions provide secondary pathways allowing the emergence of a cognitive behavior which modulates the basic abilities. In the last decades, insect's brain was deeply studied and relevant centers have been identified: Mushroom Bodies (MB) and Central Complex (CX). Even if functional details are not yet understood, it is known that they provide secondary pathways allowing the emergence of cognitive capabilities.[3]

The proposed architecture relies on these considerations and takes into account the latest results in the field of neurobiology[4] and the progress in artificial cognitive system.[5] The idea consists in producing a modulation, through a reinforcement learning, of a set of competitive and concurrent basic behaviors in order to accomplish the task assigned through a reward function. With basic behaviors, we refer to "genetically" pre-wired reflexes,

Further author information:
Send correspondence to Luca Patané, E-mail: lpatane@diees.unict.it

triggered by specific sensory events through direct sensory-motor pathways. In this work we take inspiration from the insect world and in particular from crickets emulating some of their basic skills. These are: the capability showed by crickets to recover heading during walking, called optomotor reflex; the female cricket ability to follow the sound chirp emitted by a male, named phonotaxis; and the ability to avoid obstacles, e.g. detected by the antennae. Growing up from the basic behaviors, we consider as complex behavior the ability to interpret "situations" in terms of robot-environment interaction (i.e. perception for action). The robot perceives using its sensory apparatus and processes the overall information at a cognitive level to optimize its behavior in relation to the mission assigned. The core of the architecture is constituted by the Representation layer, where different stimuli, triggering competitive reflexes, are fused to form a unique abstract picture of the environment. Each representation induces a learnable modulation of the basic behaviors in order to determine the robot overall behavior. The representation is formalized by using Turing patterns[6,7]; classical examples 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),[8] as steady state conditions. More formally, they are attractors in nonlinear dynamical systems for particular sets of environmental stimuli and serve to modulate, through a reinforcement learning, competitive and concurrent basic behaviors.

A suitable unsupervised learning, introduced at the afferent (input) stage, leads to the shaping of the basins of attractions of the Turing patterns in order to incrementally drive the association between sensor 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 the 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 as an extra-dimension by the Representation layer in order to increase the robot performances.

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 is to 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.[9] Therefore, the RD-CNN structure can be considered as the basic unit able to generate the suitable neural, self-organizing dynamics at different levels of an artificial brain architecture. It has to be outlined that several VLSI analog implementations of RD-CNNs have been developed.[10] 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, simulations in a virtual environment have been considered.

2. CONTROL ARCHITECTURE

As in insects, the proposed perceptual architecture is organized in a hierarchical structure consisting of functional blocks showing the capability to learn complex, experience-based behaviors.[11] The control architecture is described in Fig.1: it consists of series of parallel sensory-motor pathways modulated by the representation layer, working as a nonlinear feedforward complex loop, whose output is trained to combine the basic behaviors. These are pre-wired and contains the a-priori knowledge of the system. The loop is finally closed through the robot body and the environment. 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 associated emerging neural states take on the meaning of percepts. These ones are then associated to suitable modulations of the basic behaviors, driven by a Reward function. In such a way the basic behaviors, which are often life-saving sensory-motor pathways, are

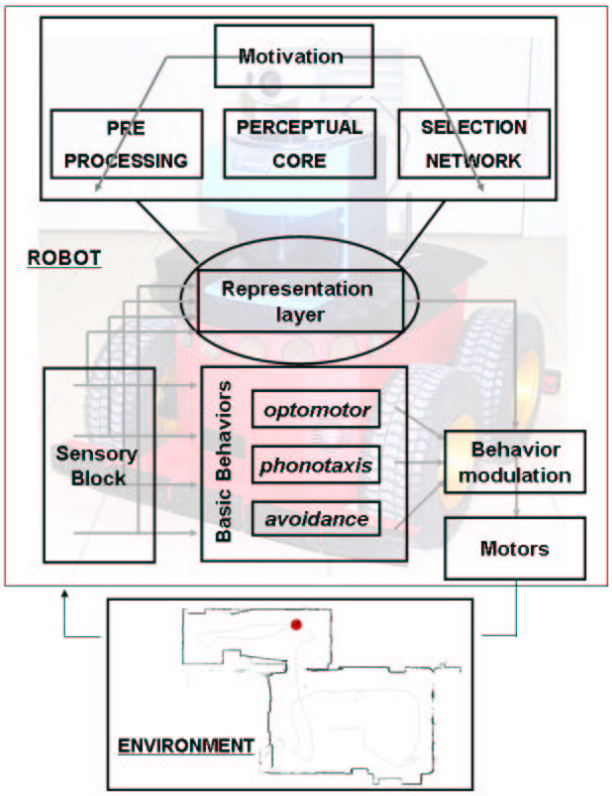


Figure 1. Functional block diagram of the implemented control architecture. The interaction between the robot and the environment is realized by direct sensory-motor pathways, the *basic behaviors*, which are modulated by the representation layer. This high level function consists of a *preprocessing block*, a *perceptual core*, a *selection network*, while the *motivation* drives the learning process.

progressively enriched with emerging capabilities which incrementally increase the animal skills. The main focus is therefore on the application of complex dynamics to obtain a proper, complex, context-dependent 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.[12] The latter in fact, uses a high level approach to face with the design of both basic behaviors and the coordination block. Here, 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. 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.[13] Turing Patterns in RD-CNN are hosted, in our architecture, within a layer here called *Representation Layer*. This term is here not referred to a place where a predictive model of the body-environment interaction is learned. This 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 organising activity, focusing at modulating the basic behaviors. Within the *Representation Layer*, a motivation-driven learning is used to associate environment conditions to internal states (i.e. Turing patterns) that modulate the system behavior to fulfill the assigned task. The RD-CNN layer leads to the emergence of a *concise representation*. In fact all the sets of environment driven multisensory information leading to one rewarding behavior modulation are collected into a unique basin of attraction, represented by its steady state condition, depicted as a pattern. This pattern is a binary image, suitable for a very compact coding.

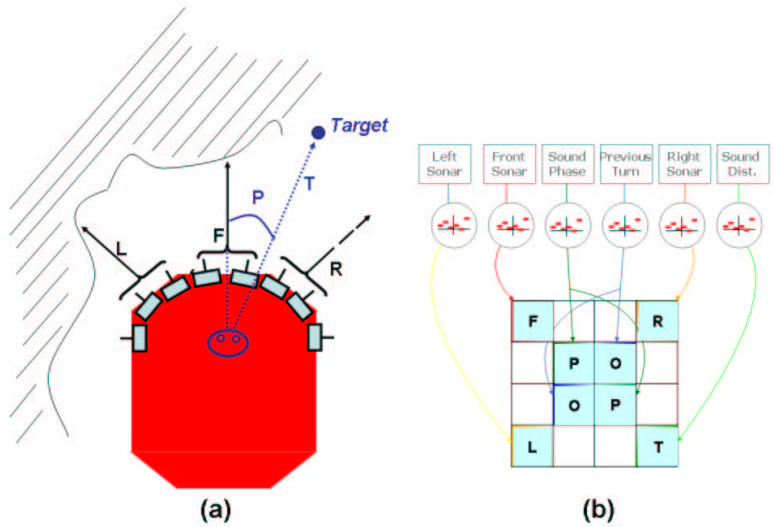


Figure 2. (a) Only six sonar sensors are used and they are arranged into three groups (F:Front, L:Left, R:Right). The target sensor provides the phase (P) and distance (T) between robot and the target. (b) Initialization for the first layer CNN cells in the representation layer. The corner cells are 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 direction robot-target.

It has to be outlined 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 4×4 network). So the number of different behavior modulations could be as large as needed to cope for very complicated and cluttered environment. 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.

2.1. Sensory block

To face with the problem of autonomous navigation, the robot is provided with eight distance sensors but only six of them are used for obstacle detection grouped into three pairs. Moreover, the robot receives information on the angle between the robot orientation and the direction 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 pre-wired reflexes, triggered by specific sensory events through direct sensory-motor pathways. Referring to crickets, these behaviours are: the capability showed by crickets to recover heading during walking, called *optomotor reflex*[14]; the female ability to follow the sound chirp emitted by a male, named *phonotaxis*[15]; 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 its leg asymmetry to maintain the heading. Even if a detailed neural network was developed to carefully model the neural control system for such behavior,[16] in this case a very simple rule was adopted consisting in: $A_o(t) = -A_F(t-1)$, 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 by 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[17] 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.[18]

2.3. Representation layer

Growing up from the basic behaviors, we consider as a *complex behavior* the ability to interpret "situations" in terms of robot-environment interaction (i.e. perception for action). 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. This 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 of 10 amplitude-varying steps learned in an unsupervised way. 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.

In this work the CNN has been designed to generate, on the basis of information coming from sensory events, Turing patterns. At the afferent level, an unsupervised learning process 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. To implement this feature, we use a nonlinear partial differential equation, discretised in space as a neural lattice of second order cells. This constitutes a two-layers RD-CNN, able to generate Turing patterns.[6] The dimension of the network has been fixed to 4×4 on the basis of a previous work.[19] Each cell $c(i, j)$ of the two-layers RD-CNN contains two state variables: ($x_{1;i,j}$ for the first layer and $x_{2;i,j}$ for the second layer, with $i, j = 1, .., 4$). The equations are:

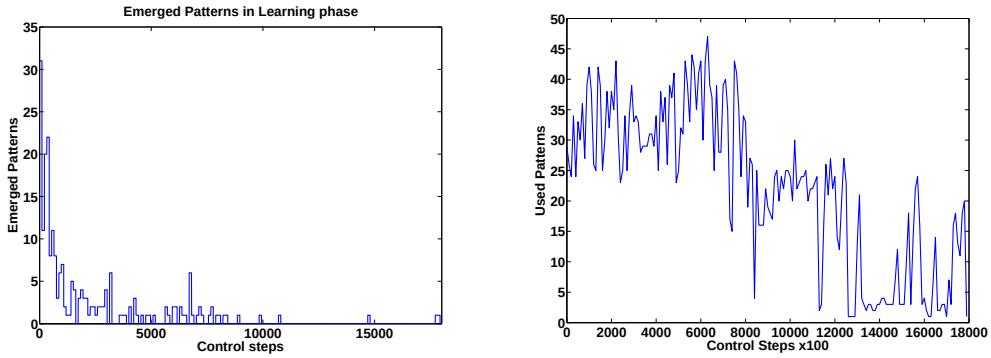
$$\begin{aligned} \dot{x}_{1;i,j} &= -x_{1;i,j} + (1 + \mu + \varepsilon)y_{1;i,j} - sy_{2;i,j} + D_1\nabla^2 x_{1;i,j} \\ \dot{x}_{2;i,j} &= -x_{2;i,j} + sy_{1;i,j} + (1 + \mu - \varepsilon)y_{2;i,j} + D_2\nabla^2 x_{2;i,j} \\ y_{h;i,j} &= \tfrac{1}{2}(|x_{h;i,j} + 1| - |x_{h;i,j} - 1|) \end{aligned}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

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: $\mu = -0.7$, $\varepsilon = 1.1$, $s = 0.9$, $D_1 = 0.05$, $D_2 = 15$, $\gamma = 1/D_1 = 20$.[19]

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.[19] The initial conditions for the state variables of the second layer are set to 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 simplify the successive processing, we associate a simple integer code for each Turing pattern as already discussed.[19] Once preprocessed the external stimuli, we reset the CNN, set the initial conditions of the selected



(a) (b)

Figure 3. (a) Number of new patterns that emerge during learning when $\gamma = 20$. (b) Number of different used patterns. The values are cumulated in windows of 100 control steps.

cells through the outputs of the SNs (Fig.2.b) and let the CNN evolve and generate a Turing pattern. Its 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*). The effect in terms of trend of new emerged patterns during learning is shown in Fig. 3.a. The number of different patterns used during the learning phase, cumulated in windows of 100 actions is shown in Fig. 3.b. A subset of the emerged patterns (about 20-30) is frequently used while the others refer to specific situations that rarely occurs.

The use of Turing patterns as steady state conditions 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 the 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 completed the learning process, at each time step t, once generated the Turing pattern $q(t)$, 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: $A_F(t) = k_o^q \cdot A_o(t) + k_a^q \cdot A_a(t) + k_p^q \cdot A_p(t)$.

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),[19,20] 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 $i = o, a, p$:

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

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 direction robot-target. The goodness of the behavior can be evaluated comparing the RF at each step by $DRF(t) = RF(t) - RF(t-1)$. A positive (negative) value for $DRF(t)$ indicates a successful (unsuccessful) behavior. Successful behaviors are followed by reinforcement, like in the Skinner's experiments[21] in order to maximize the RF. More in details, 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 to change 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. More in 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;
- the step amplitude θ_j is modified as: $\theta_i(new) = \theta_i(old) + rnd$, provided that it lies in the range $[-3, 3]$.

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, is going to be improved including the dependence on the Reward function fluctuations. More details on the whole mathematical model are given in.[19]

3. SIMULATION RESULTS

3.1. Simulation Setup

The software simulation environment, developed in $C++$, is based on the MobileSim package that allows to simulate in a 2D arena the robot Pioneer P3-AT, a commercial platform built by the Mobile Robots. This

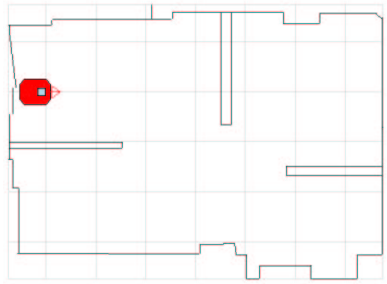


Figure 4. A screen shot of the simulator. The robot is placed in an environment divided into three rooms where a target can appear in different positions.

software tool is based on a client-server architecture and easily allows to switch between the simulated and the real robot. The arena used for the simulations consists of three rooms with six targets randomly placed (Fig. 4). The simulated environment reproduces a real environment with dimensions of about $7.5 \times 4.5\ m^2$. 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 have been used covering a range of $[-50°, +50°]$ with respect to the direction of motion. Moreover, the six sonar are grouped in three pairs (left,front and right) and the minimum distance value acquired for each pair is processed by the control architecture. The target sensor provides distance from the active target and the phase between the robot orientation and the direction robot-target. The target sensor simulates the hearing board equipped on the real robot. It is to be noticed that, for all the distance sensors, the output is saturated to the limit of the 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 direction robot-target and $A_F(t-1)$ is the rotation made by the robot in the time step $t-1$ acquired through the gyroscope. 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 $[-90°, 90°]$.

3.2. Learning phase

The task assigned to the robot consists in aiming a target avoiding obstacles. 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 conditions occurs: either the a_q averaged on the last 1000 patterns drops below 10^{-4} or 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. 5.a shows the evolution of the k_i^f for the emerged pattern 52274 and for the others (Fig. 5.b-d). It is interesting to notice how the modulation parameters that initially are uniformly distributed in their domain, due to the learning process evolve in time leading to the creation of clusters. Each cluster represents an emerged behaviour, obtained modulating the basic ones, that is successful in all the situations that are part of the basin of attraction of the corresponding pattern.

3.3. 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

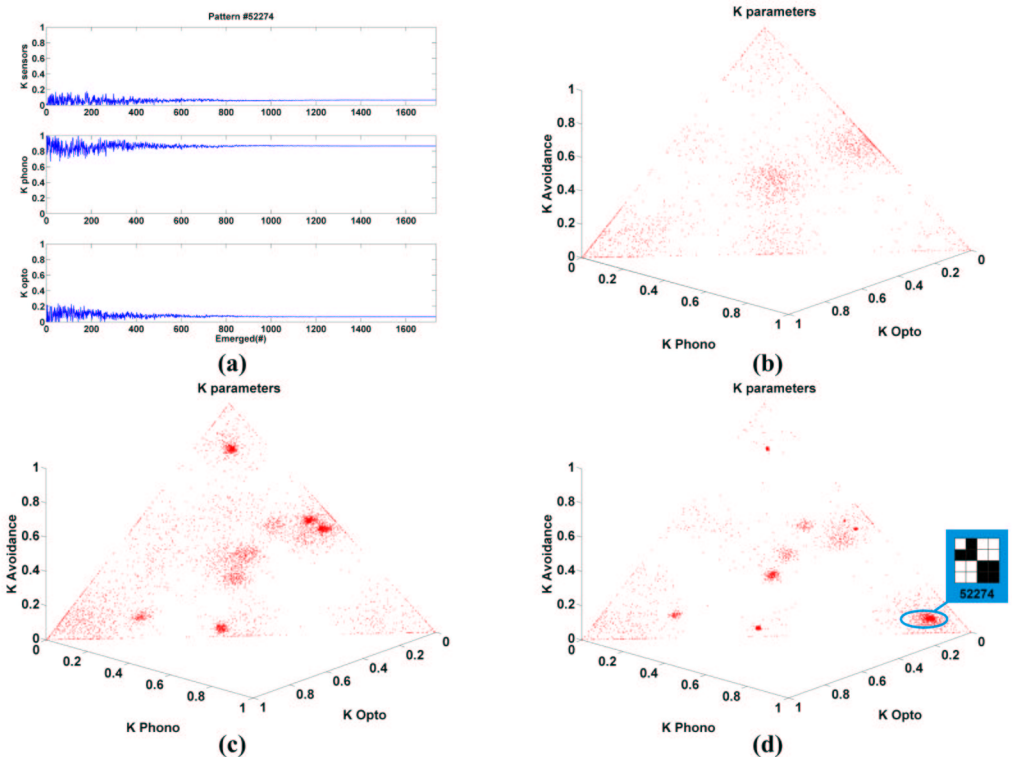


Figure 5. (a) Evolution of the k_i^f for the emerged pattern (52274). The evolution of the modulation parameters is shown in (b-d) where the solutions adopted at the beginning of the learning stage (b), between 5000 and 10000 movements (c) and for the last 5000 movements are shown. In the last picture it is also indicated the region associated with the pattern 52274.

parameters. The constant behavior modulation parameters were chosen through a manual tuning aiming at optimize 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 behaviour of the robot at the beginning of the learning phase when the behaviour modulation is initialized with random values.

For the performance validation four different sequences of activation for the targets have been used. Fig. 6 shows examples of trajectories followed during the testing phase in case of fixed, random and learned modulation parameters for different target searching sequences. The compared results, in terms of number of steps to complete the sequence, and number of collisions that occur, are reported in Tab. 1 and in Tab. 2.

The learning process leads to a significant reduction of the actions needed to complete a sequence of target searching, demonstrating the effectiveness of the control architecture and its capability to generalize the representations.

Videos and multimedia material are available on the web.[22]

4. REMARKS AND CONCLUSIONS

In this paper a new control architecture for the action-oriented perception in roving robots is described and validated through simulations referring to a foraging task. The control architecture is based on some pre-wired basic behaviors which are modulated by the Representation Layer. This layer learns to associate a set of sensory events with specific Turing patterns and with modulation parameters that affect the basic behaviors. We used complex dynamics and attractor based nonlinear computation together with 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

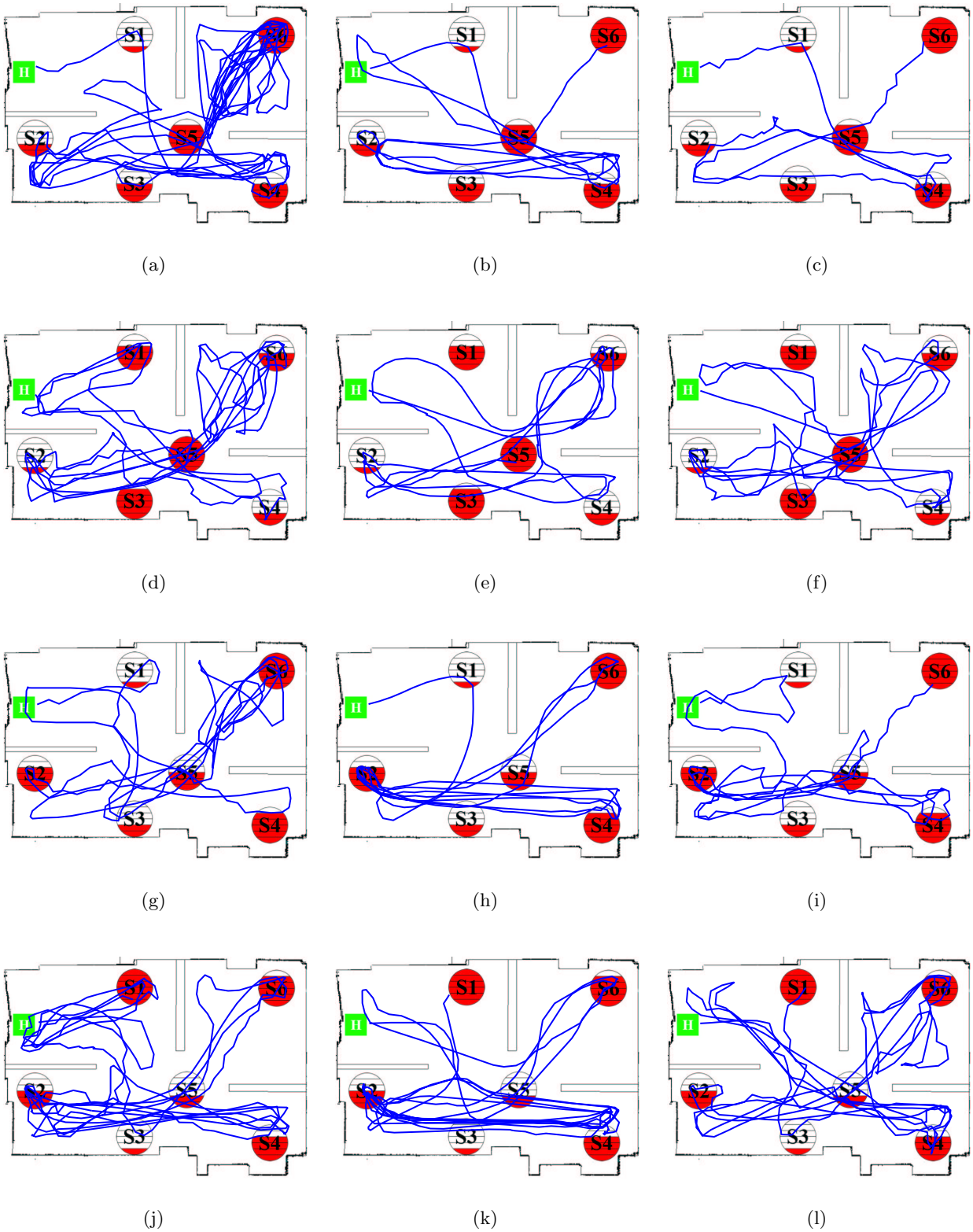


Figure 6. Trajectories followed by the robot controlled through: random (a)(d)(g)(j), fixed (b)(e)(h)(k) and learned (c)(f)(i)(l) parameters for the sequence P0 (a)(b)(c), P1 (d)(e)(f), P2 (g)(h)(i) and P3 (j)(k)(l). H is the starting point and the sequences are: P0 (S1-S2-S3-S4-S5-S6); P1 (S2-S4-S6-S1-S3-S5); P2 (S1-S3-S5-S2-S4-S6); P3 (S3-S5-S2-S4-S6-S1)

Table 1. 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 sequence 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 2. 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 Bumps			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

relevant details about the robot body and position in the environment are naturally used to achieve an efficient, embodied and situated knowledge. It is to be underlined that algorithms dedicated to face with 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 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. The implementation of the whole architecture on board on the robot in view of a more complete and autonomous interaction with complex and cluttered environments is also envisaged.

ACKNOWLEDGMENTS

The authors acknowledge the support of the European Commission under the project SPARK II "Spatial-temporal patterns for action-oriented perception in roving robots: an insect brain computational model", EU Project SPARK II FP7-ICT-2007-1-216227.

REFERENCES

1. R. A. Brooks, "Intelligence without reason", *in 12th Int. Conference on Artificial Intelligence*, eds. J. Mylopoulos and R. Reiter, San Mateo, CA; 1994.
2. S. Nolfi, "Power and Limits of Reactive Agents", *Neurocomputing*, vol. 42, no.1, pp. 119-145, 2002.
3. B. Webb and J. Wessnitzer Perception for action in insects, In P. Arena and L. Patan (eds), "Spatial Temporal Patterns for Action Oriented Perception in Roving Robots", Cognitive Systems Monographs Vol. 1, Springer (Feb. 2009), ISBN: 978-3-540-88463-7.
4. W.J. Freeman, "How and why brains create meaning from sensory information", *Int. Jour. of Bif. and Chaos*, vol.14, pp.515-530, 2004.
5. D. Vernon, G. Metta and G. Sandini, "A Survey of Artificial Cognitive System: Implications for the Autonomous Development of Mental Capabilities in Computational Agents", *IEEE Transections on Evolutionary Computation*, vol. 11, no.2, pp. 151-180, 2007.

6. A. M. Turing, "The chemical basis of morphogenesis", *Phil. Trans. Roy. Soc. Lond. B*, vol. 237, pp. 37-72, 1952.
7. J. D. Murray, *Mathematical Biology I: An Introduction (third edition)*, Springer-Verlag, NJ; 2002.
8. L. Goras and L. Chua, "Turing Patterns in CNNs - Part I, II. *IEEE Trans. Circuits and Systems - I*, vol. 42, pp. 602-626, 1995.
9. P. Arena, L. Fortuna, M. Branciforte, "Reaction-Diffusion CNN Algorithms to Generate and Control Artificial Locomotion", *IEEE Trans. on Circuits and Systems, I*, vol. 46(2), pp.259-266, 1999.
10. P. Arena, L. Fortuna, M. Frasca, L. Patané, "A CNN-based chip for robot locomotion control", *IEEE Trans. On Circuits and Systems-I*, Vol. 52, No. 9, pp. 1862-1871, 2005.
11. J. Wessnitzer, B. Webb, "Multimodal sensory integration in insects - towards insect brain control architectures", *Bioinspiration and Biomimetics* vol. 1, pp. 63-75, 2006.
12. R.A. Brooks, "A robust layered control system for a mobile robot", *IEEE Jour. of Robotics and Automation*, Vol. 2, no. 1 pp. 14- 23, 1986.
13. R. C. Arkin, *Behaviour Based Robotics*, MIT Press, Cambridge, 1991.
14. H. Böhm, K. Schildberger, F. Huber, "Visual and acoustic course control in the cricket Gryllus-bimaculatus", *Journal of Experimental Biology* vol. 159, pp. 235-248, 1991.
15. B. Webb and T. Scutt, "A simple latency dependent spiking neuron model of cricket phonotaxis", *Biological Cybernetics*, vol. 82(3), pp. 247-269, 2000.
16. P. Russo, B. Webb, R. Reeve, P. Arena, L. Patané, "A Cricket-Inspired Neural Network for FeedForward Compensation and Multisensory Integration", *in IEEE Conference on Decision and Control and European Control Conference 2005*, Sevilla, Spain, 2005.
17. J. Borenstein and Y. Koren, "The vector field histogram fast obstacle avoidance for mobile robots", *IEEE Journal of Robotics and Automation*, vol. 7, pp 278-288, 1991.
18. A. Horchler, R. Reeve, B. Webb and R. Quinn, "Robot phonotaxis in the wild: a biologically inspired approach to outdoor sound localization", *Advanced Robotics*, vol.18, no.8, pp. 801-816, 2004.
19. P. Arena, P. Crucitti, L. Fortuna, M. Frasca, D. Lombardo and L. Patané, "Turing patterns in RD-CNNs for the emergence of perceptual states in roving robots", *Int. Journal of Bifurcation and Chaos*, vol. 17, no.1, pp. 107-127, 2007.
20. K. Schulten, "Theoretical biophysics of living systems". In H. Ritter, T. Martinetz and K. Schulten (Eds.), *Neural computation and self-organizing maps: An introduction*. Addison-Wesley, New York; 1992.
21. B. F. Skinner, *About behaviorism*, Alfred Knopf, New York; 1974.
22. EU Project SPARK II, online at *www.spark2.diees.unict.it*.