CNN and collective perception

P. Arena, L. Fortuna, D. Lombardo, L. Patané
DIEES - Dipartimento di Ingegneria Elettrica, Elettronica e dei Sistemi
Facoltá di Ingegneria - Universitá degli Studi di Catania viale A. Doria 6,
95125 Catania, Italy
Email: lpatane@diees.unict.it

***Abstract*— Cooperation is an ability used by particular animal species (e.g. ants) to survive in the world. In the bio-inspired robotics field, several attempts were carried out to exploit the application of cooperative strategies both to accomplish tasks otherwise impossible and to improve the global performances of the group of robots. In this paper we discuss on the emergence of collective behaviours, in groups of robots individually controlled by a cognitive architecture based on Turing Patterns. Robots can exchange information each other when specific situations are encountered: deadlock and target retrieving. This information spreading mechanism reduces the learning time and increases the final performance of the group.**

I. Introduction

During the last years a considerable attention was devoted to investigate the applicability of animal cooperative behaviours in the field of mobile robotics. Nowadays, international projects have been activated to face with this problem from different prospectives, with the common aim to reach cooperation among autonomous robots to fulfil a given task.

To have an idea about the interest of the scientific community on cooperative strategies, we can mention some of the ongoing activities that approach this topic from different prospectives proposing interesting applications. The JAST Project (Joint-Action Science and Technology) is focused on building jointly-acting autonomous systems that communicate and work intelligently on mutual tasks, think to a group of manipulators that cooperate in a manufacturing process. The main objective of the GUARDIANS Project (Group of Unmanned Assistant Robots Deployed In Aggregative Navigation supported by Scent detection) is to develop a swarm of autonomous robots used to explore and search for targets in an urban ground. The IWARD Project (Intelligent Robot Swarm for Attendance, Recognition, Cleaning and Delivery) extends the principles of robot cooperation in hospitals and healthcare centres to overcome the shortages of healthcare staff. More information can be found in the EU Cognitive Systems research activity program [1].

All these projects try to propose new architectures and learning strategies to reach the overall goal. One of the paradigms largely used is based on evolutionary computation. Several works on this topic, describe a scenario in which agents learn by evolution how to handle their sensors/actuators and how to cooperate with the rest of sub-agents [7]. Emergence of behaviors happens when the co-evolution of several sub-agents converges towards an architecture suitable for the assigned task. In other words, evolutionary algorithms (e.g. genetic algorithms) can be used to specialize robots into accomplish difficult tasks that are often inspired by the insect world. An example is the Ant Colony Optimization (ACO), a methodology completely inspired by the ant's capabilities. Ants use with cooperation to solve problems otherwise impossible [2], [3].

Besides computer simulations that are the starting point of this research, real robots have been designed and tested. For instance the Swarm-bots, aimed to study new approaches to the design and implementation of self-organizing and self-assembling artifacts [4]. Key aspects that concur in designing an architecture for cooperation are the choice of sensors to be embedded in the robot and the interaction mechanisms that are at the basis of each cooperative behaviour. Interaction can be formalized at different level of complexity as a communication language that through different forms can be used to spread information, assign different tasks to each robot of the team, and so on.

This paper is located inside this fervent research activity. The results of the SPARK Project [5] have been used as starting point for this work. The main aim of the SPARK project activities was the formulation of a cognitive model, inspired by the insect brain architecture, consisting of a hierarchical structure with parallel pathways. The core of the system is the representation layer that starting from the environmental perception, creates a concise portrait of the situation by means of a Turing Pattern. It is used to modulate the system behaviour that is guided by a reward function. The learning mechanisms act at the afferent layer, to model the basins of attraction of the CNN-based dynamical system generating patterns and at the efferent layer where the meaning of the pattern is created on the basis of the mission assigned to the robot [6]. Autonomous robots, interacting with the environment, acquire information that can be used to solve an assigned task. Each robot, to improve its performance, can cooperate with the others in solving the task.

The cognitive architecture was applied both in simulated and in real roving robots as discussed in [8]. In this paper we want to demonstrate how a group of robots, each one equipped with an independent control system, can show collective behaviours when a simple communication mechanism is activated.

In the next section the basic cognitive architecture is introduced, section III describes the communication framework that constitutes the basic mechanism for the emergence of collective behaviours. Simulation results are given in section

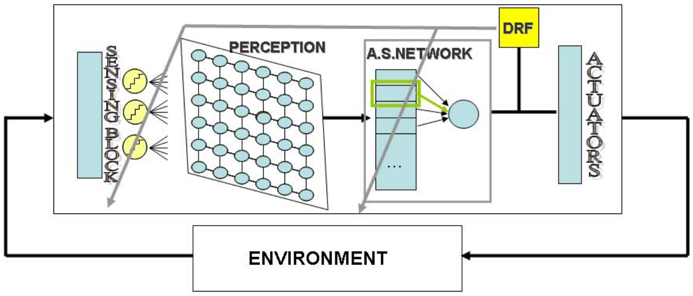


Fig. 1. Scheme of the interaction between the robot and the environment: the block representing the agent contain the whole control architecture. The sensory events are preprocessed in the *Sensing Block* before enter the perceptual block (*Perception* in figure), which forms an abstract representation of the current state of the environment. The association between this and the actual action to be performed is made by an *Action Selection Network*. Two learning algorithms act to build up the associations, at the afferent and at the efferent layer, driven by a reward block, called *Difference Reward Function Block* (DRF in figure, see text for more details).

IV and finally conclusions are drawn in section V.

II. Cognitive architecture

The whole perceptual process has the aim to transform different environmental situations into representations, triggering specific actions. These mechanisms are plastically modified by experience. A scheme of the action-oriented perception process is shown in Fig. 1.

The dynamic processing of external stimuli is at the basis of the creation of a concise representation of the environment, fundamental aspect for the perceptual process. To implement this feature, a nonlinear dynamical system has been considered: a two-layer 4×4 RD-CNN, able to generate Turing patterns [9]. The dimension of the network has been fixed to 4×4 on the basis of other previous works [6].

The system equation are following reported:

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

where $y_{h;i,j}$ $(h = 1, 2)$ is the output of the layer l of the cell (i, j) and D_1, D_2, μ, ε and s are parameters of the model.
To obtain the emergence of Turing patterns, it is necessary that some of the spatial modes, related to the chosen geometry, are within a “Band of unstable modes” (Bu), i.e. correspond to positive temporal eigenvalues. The conditions to obtain Turing patterns are [11]:

$$\begin{cases} \mu < 0 \\ \mu^2 + s^2 > \varepsilon^2 \\ \varepsilon > -\mu \frac{d+1}{d-1} \\ \frac{[d(\mu+\varepsilon)+(\mu-\varepsilon)]^2}{4d} > \mu^2 - \varepsilon^2 + s^2 \end{cases} \tag{2}$$

where $d = \frac{D_2}{D_1}$. To satisfy the above discussed conditions, the chosen parameters have been $\mu = -0.7$, $\varepsilon = 1.1$, $s = 0.9$, $D_1 = 0.05$, $D_2 = 15$.
The output of the sensing block sets the initial conditions for the first layer state variable of a subset of cells; all the other initial conditions are set to zero. The boundary conditions are set as zero-flux.
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 above 1 or below -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 following processing, we associate a simple integer code for each Turing pattern:

1) the first-layer cells are enumerated starting from the upper left-hand corner.
2) a symbolic value $y_{simb,c}$ is associated with each cell c as follows:
   - if the cell output is $y_{1,c} = -1$ then $y_{simb,c} = 0$
   - if the cell output is $y_{1,c} = 1$ then $y_{simb,c} = 1$
3) an integer code is associated with the steady-state pattern:

$$code = \sum_{c=0}^{15} y_{simb,c} 2^c \tag{3}$$

To finely represent different environmental situations, it would be feasible to have a large number of Turing patterns. A trade-off between the amount of different patterns and easiness of control has to be found. The number of the emerging Turing patterns can be modified tuning the parameter $\gamma = \frac{1}{D_1}$: by increasing such parameter, the number of possible modes to be selected increases consequently [6]. In this way, a stronger competition between the allowed modes is triggered to generate patterns. In this paper, we set the parameter $\gamma = 20$.
Once processed the external stimuli, we reset the CNN, set the initial conditions through the outputs of the sensing neurons 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 Pattern Vector can contain up to 200 items: this number could seem to be very large, but in the first part of the learning phase, the number of emerged patterns is very high, so the Pattern Vector should be large enough to avoid that the frequent substitution of patterns in the table would cause the loss of the learning information. If the pattern vector is full, the new element replaces the least recently used (LRU), i.e. that one with the lowest *Occurrence Lag* value.
The use of the steady states of a dynamical system implies a form of sensory fusion, i.e. we synthesize lots of heterogeneous sensor information into a single attractor. At each step, the information coming from sensors are fused to form a unique abstract representation of the environment.

A. *The Selection Network and the motivation layer*

The Action Selection Network associates each element q of the pattern vector with an action A_q. An action consists of two elements, the module and the phase of the robot movement setting, respectively, the translational step and the rotation. Each element q of the pattern vector is connected with two weights, $w_{q,m}$ and $w_{q,p}$, representing, respectively, module and phase of the action A_q. In this paper, we keep fixed the weight $w_{q,m} = w_m$ for all the patterns and vary only the $w_{q,p}$ through a reward-based reinforcement learning implemented by a simplified Motor Map (MM) [10], [12]. Emulating the associative learning in animals, we determine the goodness of an action by means of a Reward Function (RF) defined as follows:

$$RF = -\sum_i k_i \cdot f_i(e^{d_i}) - h \cdot f_T(|\phi_T|) \tag{4}$$

where d_i is the distance between the robot and the obstacle detected by the sensor i ($i = Front(F), Right(R), Left(L)$), Φ_T is the angle between the robot orientation and the direction connecting robot and target, while k_i and h are positive constants determined in a design phase. The functions $f_i(\cdot)$ and $f_T(\cdot)$ are used to scale all the terms in the reward function. The RF indicates the mission assigned to the robot: in this way it knows *what to do* but not *how to do* that. The RF is used to learn the sensing-perception-action cycle by directly driving the association between Turing patterns and actions and indirectly modulating the basins of attraction of the Turing patterns. This modulation is realized by training the sensory layer. In fact the RD-CNN is exploited for it capability to rapidly converge to steady state patterns, with associated basins of attraction. These ones typically depend on the CNN parameters, and so are fixed. In order to add plasticity to them, we act on the afferent association between the stimuli and the CNN itself, by means of a simple associative learning aimed to establish the correct association between the sensory events and the internal representations. The scope of the learning algorithm is to maximize the RF: small absolute values in (4) indicate good situations for the robot. The goodness of an action performed at the step t, is provided by $DRF(t) = RF(t) - RF(t-1)$. A positive (negative) value for DRF indicates a successful (unsuccessful) action. Successful actions are followed by reinforcement, like in the Skinner's experiments [13].

B. The Sensing Block

The Sensing Neurons (SNs) are responsible for transforming the incoming stimuli into initial conditions for the RD-CNN which will converge to a Turing pattern. Our choice for the SNs activation function consists in an increasing function constituted of ten variable amplitude steps, θ_i ($1 \le i \le 10$), covering the whole input range $[-1, 1]$. At the beginning of the learning phase, all the steps are initialized to uniformly cover the codomain $[-1.2, 1.2]$. At each step, if the performed action has positive effects ($DRF > 0$), then the step amplitude does not change. Otherwise, when the action results to be negative, the step amplitudes are modified randomly (in order to modulate the basins of attraction for the patterns). 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 (situation) 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 positive actions. The system constituted by the RD-CNN plus the SNs with simple learning can be seen as a unique perceptual system where the pool of neurons within the CNN generates patterns, as steady state solutions of neural lattices. The SNs learn to suitably associate set of environmental conditions to a given pattern.

More in detail, if the action associated with the currently emerged pattern is unsuccessful (i.e. $DRF < 0$), then the learning algorithm for each SN acts as follows:

- determine which of the RF component 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;
- if rnd is positive, the ten step amplitudes θ_j are modified as:

$$\begin{cases} \theta_j(new) = \theta_j(old) & if \;\; j < i \\ \theta_j(new) = \theta_j(old) + rnd & if \;\; j \ge i \end{cases} \tag{5}$$

- instead, if rnd is negative:

$$\begin{cases} \theta_j(new) = \theta_j(old) - |rnd| & if \;\; j \le i \\ \theta_j(new) = \theta_j(old) & if \;\; j > i \end{cases} \tag{6}$$

- the step function is saturated at $[-1.2, 1.2]$.

To guarantee the convergence of the algorithm, the variable r varies in the range [-h,h] where h, initially sets to 0.5, decreases at each step with an aging coefficient $h_{new} = 0.999 \cdot h_{old}$. 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 in analogy with the most recent studies and related hypotheses in neurobiology [14], which show how the process of learning situations is obtained modulating the basins of attraction in the dynamical state space of the brain.

More details on the whole mathematical model are given in [6].

III. Communication framework

To improve the performance obtained with a single robot, the problem of interaction in a group of robots has been considered. The basic idea consists in realizing a communication layer used to exchange the successful behavioural scheme learned by each robot. Each robot has an input and an output buffer, used to exchange information in terms of sensorial stimuli and corresponding actions. The final aim is to transform the knowledge acquired by a single robot to the group mimicking a reaction-diffusion process. Two specific situations have been considered as triggering events to activate cooperation: deadlock and target found.

As far as a deadlock situation is concerned, the trapped robot looks for another member of the group randomly selected. The sensory status of the deadlocked agent is transferred to the other that processes this information like in the case it will be in that conditions. The motor response that arises is then given to the trapped robot that performs this action and then evaluates its new condition in terms of reward function, if the robot obtains an improvement, the action is stored overwriting the old one.

When a robot retrieves a target, it looks for all the robots present in its neighborhood that have collected a lower number of targets. To improve their retrieving capability, it sends the last ten actions performed to the others that store the corresponding perception-action association in their perceptual system. This communication mechanism improves the robot behaviour in proximity of a target increasing its collecting capability.

After each information exchange, the robot that receives the data, is inhibited (i.e. is not able to communicate) for a given time. This refractory period has been introduced to avoid a continuous exchange of data that can destabilize a robot that instead have to perform its own experience to learn by itself and to give its contribution to the group.

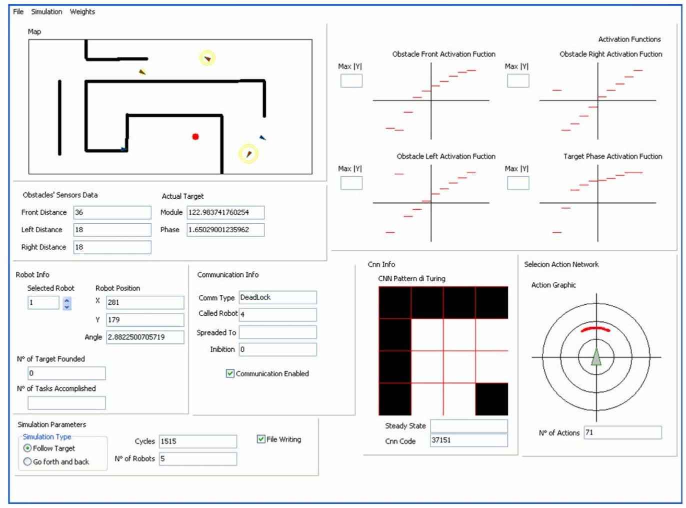


Fig. 2. Simulation environment. For each robot it is possible to visualize the sensory neurons, the sequence of emerged patterns, the action selection network and the communication events.

IV. Simulation results

The proposed architecture was tested in a simulation environment in which a group of simulated roving robots was requested to perform a food retrieval task.

A. Simulation environment

The software simulation environment, developed in $C++$, allows to create an arena constituted by walls, obstacles and targets. Moreover, in the arena a series of robots equipped with a distributed sensory system can be simulated. The software interface is shown in Fig. 2. An arbitrary number of robots can be simulated in a given arena, and for the selected robot a number of useful information are given: the shape of the sensory neurons updated during the learning process, the sequence of patterns that emerges, the corresponding action associated through the Action Selection Network and the communication status.

The arena, used for the simulations, has dimension of $440 \times 210\ pixels^2$ and it is filled by obstacles and a target source. Each robot is equipped with three distance and one target sensors. The front side sensor detects obstacles within a limited range of 40 pixels in a visual field of $[-45°, 45°]$ with respect to the robot longitudinal axis. The other two obstacle sensors have visual field of $[-30°, 30°]$ with respect to the direction orthogonal to each robot side and a detection range of 20 pixels. 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 40 pixels for the front distance sensor, and 20 pixels for the other two distance sensors. The target sensor has an unlimited range and provides the angle between the robot orientation and the robot-target direction. All the sensor outputs are scaled in the range $[-1, 1]$. The scaled sensors are used as inputs for the SNs and the output of this operation is finally assigned as initial condition for the CNN corner cells. In the simulations here proposed, the front, left and right distance sensors are associated to the cells C(1,1), C(4,1) and C(1,4) respectively whereas th target sensor is used to determine the initial condition for the cell C(4,4).

The component of the RF in eq.(4) are defined as:

- $f_F(e^{d_F}) = -(1/(e^{\gamma_F(d_F+1)})$
- $f_L(e^{d_L}) = -(1/(e^{\gamma_L(d_L+1)})$
- $f_R(e^{d_R}) = -(1/(e^{\gamma_R(d_R+1)})$
- $f_T(\Phi_T) = -|\Phi_T|$

where $\gamma_F = 2$, $\gamma_L = \gamma_R = 3$ and d_F, d_R, d_L are the scaled distances detected by the sensors. In the following simulations, the choice for the other parameters in (4) was: $k_F = 70$, $k_L = k_R = 40$ and $h = 20$. 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 $[-20°, 20°]$.

B. Swarm behaviour

The task assigned to the robot consists in reaching a target avoiding obstacles. The learning phase is performed for 10000 cycles (i.e. actions). The arena used during the training phase consists of sparse obstacles randomly distributed and a target source (see Fig. 3). After reaching the current active target, the robot is freezed for a given time in order to allow the other robots to complete the mission. When all the robots have reached the target or after the waiting period, a new target is randomly placed in the arena substituting the old one, that is turned off, and all the robots will be attracted. In the following simulations, the waiting time has been fixed to 300 steps.

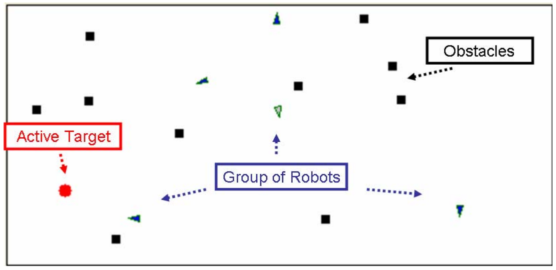


Fig. 3. Arena used for the simulations.

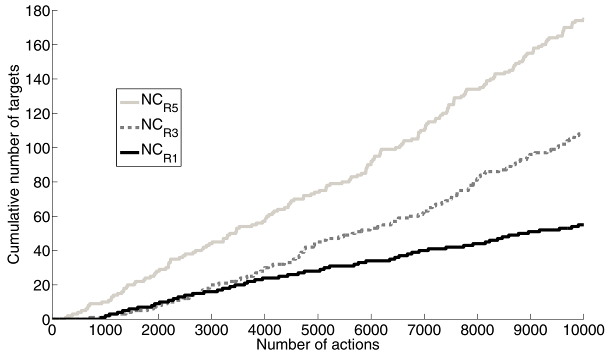
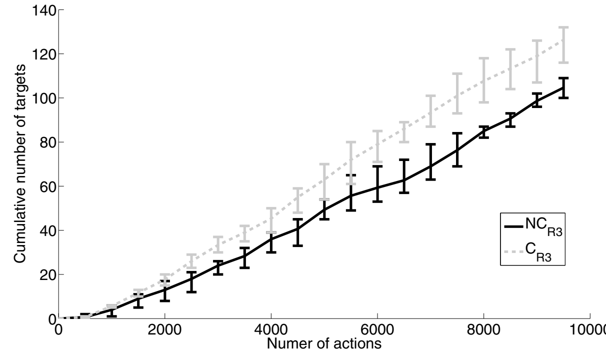


Fig. 4. Comparison between the cumulative number of targets collected by groups of one, three, and five robots when communication mechanisms are disabled.

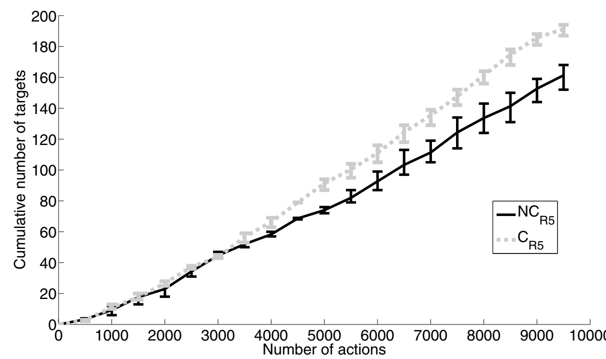
At the beginning of the learning phase, the robot performs random actions due to the random initialization of the phase weights $w_{p,q}$, which determine the robot heading. During the learning process, the Motor Map-like algorithm corrects the action associated with each pattern.

The foraging task can also be solved increasing the number of robots that concurrently navigate in the environment collecting targets. It is evident that multiple robots perform better than a single one and the cumulative number of retrieved targets is related to the number of robots. Fig. 4 shows the performance comparison in the case of one, three and five robots when communication mechanisms are turned off. When the communication layer is introduced, an evident speed-up in the learning process is obtained due to the possibility to exchange information among the group. Taking into account, also in this case, as performance index the cumulative number of targets retrieved by the group, the improvement in case of three and five robots is shown in Fig. 5 where the average value of the performance index over five simulations is reported. From this preliminary analysis, an improvement of about 20% has been obtained.

We want to underline how simple communication rules introduced as a further layer (i.e. communication layer) to connect independent perceptual architectures that evolves autonomously, improves the group capability to accomplish the assigned task. Moreover the information exchange lead to the emergence of collective behaviours. In Fig. 6, an example of the trajectory followed by a group of three robots is shown.



(a)



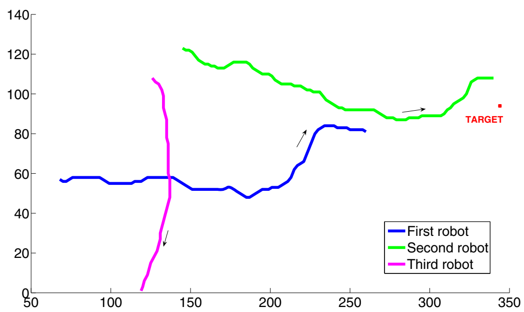
(b)

Fig. 5. Improvement obtained with the introduction of communication, in terms of number of target retrieved during learning, in case of a group of three (a) and five (b) robots. The results represent the number of targets retrieved by the group and cumulated during time (i.e. number of actions). The graph shows the average value obtained in a set of five simulations and the bars indicate the minimum and maximum difference among the simulations. NC:communication disabled, C:communication enable.

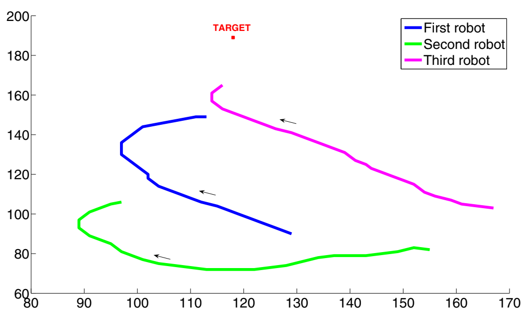
At the beginning of the learning process (see Fig. 6 (a)), the robot behaviours are completely unrelated each other. During learning the robots transfer among the group successful relations between perception and actions, this process allows, in some cases, the emergence of collective behaviours like flocking (see Fig. 6 (b)). As demonstrated by these preliminary results, the proposed perceptual architecture has been extended to a group of robots, trying to accomplish a common task, and the global performances can be improved introducing a communication layer.

V. Conclusions

The emergence of collective behaviours in a group of robots is an important aspect inside the research field of advanced robotics and cognitive systems. In this paper we report the simulation results obtained when a cognitive architecture based on Turing pattern generation, is applied to control each robot of a team performing a foraging task. The results show how



(a)



(b)

Fig. 6. Emergence of collective behaviours. (a) At the beginning of the simulation, the robot trajectories are mostly independent each other, (b) after 8000 actions, due to the communication mechanisms, a flocking behaviour emerges.

the introduction of a communication layer increase the performance in the team. The communication occurs only in case of highly rewarding events (i.e. target detection) or punishing events (i.e. deadlock situations). Further works will include the extension of this approach to different tasks comparing the results with other cooperation strategies. Other interesting scenarios could include cooperation in garbage collection or specialization of the robot behaviours in order to complete a mission structured in sub-tasks. Another interesting aspect that will be investigated consists in allowing the agent interaction not through direct communication but creating a link through the reward function. In this case the robots should learn to behave in different way, guided by a global reward function that takes into account the contribution given to the mission by each robot. Finally an hardware implementation is envisaged and experiments with real robots in structured and unstructured environments will be carried out.

ACKNOWLEDGMENT

This work was supported the European Commission under the project FP7-INFSO-ICT-216227 SPARK II "Spatial-temporal patterns for action-oriented perception in roving robots II: an insect brain computational model".

REFERENCES

[1] IST Cognitive Systems Web page [available on-line]: $http://cordis.europa.eu/ist/cognition/projects.htm$
[2] M. Dorigo and T. Stützle, *Ant Colony Optimization*, MIT Press, 2004.
[3] M. Dorigo, C. Blum, "Ant colony optimization theory: A survey". *Theor. Comput. Sci.* 344(2-3): 243-278, 2005.
[4] Groß R., Bonani M., Mondada F., Dorigo M., "Autonomous Self-Assembly in Swarm-Bots", in *IEEE Transactions on Robotics*, vol. 22(6), 2006.
[5] SPARK Web page [available on-line]: www.spark.diees.unict.it
[6] Arena P., Crucitti P., Fortuna L., Frasca M., Lombardo D., Patané L., "Turing patterns in RD-CNNs for the emergence of perceptual states in roving robots", *International Journal of Bifurcation and Chaos*, 18(1), pp. 107-127, 2007.
[7] S. Nolfi and D. Floreano *Evolutionary Robotics The Biology, Intelligence, and Technology of Self-Organizing Machines*, MIT Press, 2000.
[8] P. Arena, L. Fortuna, D. Lombardo, L. Patané, "Perception for Action: Dynamic Spatiotemporal Patterns Applied on a Roving Robot", to appear in Adaptive Behaviour, DOI: 10.1177/1059712308089181, 2008.
[9] Turing, A. M., The chemical basis of morphogenesis. *Phil. Trans. Roy. Soc. Lond. B, 237*, 37-72, 1952.
[10] Arena, P., Fortuna, L., Frasca, M., & Sicurella, G., An adaptive, self-organizing dynamical system for hierarchical control of bio-inspired locomotion. *IEEE Transactions on Systems, Man and Cybernetics, Part B, 34(4)*, 1823-1837, 2004.
[11] Murray, J. D., *Mathematical Biology I: An Introduction (third edition)*, New York:Springer-Verlag, 2002.
[12] Schulten, K., *Theoretical biophysics of living systems.* In H. Ritter, T. Martinetz, and K. Schulten (Eds.), Neural computation and self-organizing maps: An introduction. New York: Addison-Wesley, 1992.
[13] Skinner, B. F., *About behaviorism*, New York:Alfred Knopf, 1974.
[14] Freeman, W. J., How and why brains create meaning from sensory information. *International Journal of Bifurcation and Chaos, 14*, 515-530, 2004.