Pergamon

Robotics & Computer-Integrated Manufacturing, Vol. 11, No. 4, pp. 279–286, 1994

0736-5845/94 $7.00 + 0.00

0736-5845(95)00004-6

● *Paper*

AVOIDING THE WORLD MODEL TRAP: AN ACTING ROBOT DOES NOT NEED TO BE SO SMART!

PHILIPPE GAUSSIER* and STEPHANE ZREHEN*

*ENSEA ETIS, 6 av du Ponceau, 95014 Cergy Cedex, France
*LAMI, EPFL-DI, CH-1015 Lausanne, Switzerland

We propose several examples of robotics applications where interactions with the environment play an important role, and where no modelization of the robot or of the world is needed. We argue that, in general, this helps to simplify the complexity of the control system, and illustrate this claim through several examples. We propose different neural networks which allow clustering of scattered objects by several robots, learning of obstacle avoidance and learning of target retrieval.

1. INTRODUCTION

Today, robotics systems such as arm manipulators or sorting systems are very efficient for particular applications. For instance, an industrial mobile robot can go from one point to another to transport objects by following a line on the floor or using artificial landmarks such as reflectors. The success of this kind of approach lies in the ability of engineers to modify the human environment to simplify the robot's work. However, if one wishes to design a robot suited to an unknown environment or to different applications this method cannot be applied. Our general goal is to design control architectures for autonomous robots capable of adapting to an unknown world. Our theoretical framework is the constructivist approach,[12,15] which can be more easily implemented by a "subsumption-like" methodology.[3] We start by trying to solve a simple problem for a robot in an unknown world. This leads us to draw our inspiration from biological systems. We try to bring together different results from neurobiology, ethology and psychology to build neural-like control architecture which allows the emergence of interesting observed behaviours such as collective clustering, scene recognition and navigation. In accordance with constructivism, our main claim is that acting in the world is necessary to the interpretation of perceived signal, i.e. to the emergence of a "cognition". In this paper, we will emphasize the interest of a constructive approach: we build one level, and when the resulting behaviour is well known, we add other blocks to improve the robot's performances and to solve more complex tasks. If an interesting emergent behaviour occurs at a lower level, we can amplify it directly at a higher level without programming it completely. We thus end up with simpler programs than in a top down programming approach. Moreover, we will show how performing an action modifies the subsequent perceptions (thus reducing the complexity of the recognizable scenes) and how actions simplify "reasoning programs".

Obviously, we do not claim that this approach will have good performances for such tasks as airplane booking, or any well-formalized applications for which optimized algorithms have already been proposed. We hope to imagine a single system that can be reasonably efficient in different domains. In order to illustrate this concept we present in this paper different examples of robot applications of increasing complexity that use the same type of neural architecture, based on the PerAc (Perception–Action) building block. These examples also demonstrate how to take into account the robot's actions and why it is useful for the design of autonomous systems. In the first part, we present a robot that learns to avoid collisions with obstacles while wandering in an arena, thereby learning an appropriate representation of its body shape to avoid obstacles. Then, we propose an experiment of object clustering by cooperative robots to illustrate the previous points and to point out the interest of competitive mechanisms. Finally, we present a robot that uses visual information to learn how to retrieve a goal in an unknown environment, using the PerAc architecture. There we will focus on the fact that action implies doing only one thing at a time, which simplifies the robot's reasoning.

Acknowledgements—This research is supported by the Swiss National Fund's NRP 23 program. We would like to thank Francesco Mondada for designing the robot, and for his precious help during this research, and the French Ministry of Foreign Affairs who allowed P. Gaussier to make his civil service at the LAMI.

2. ACTING TO LEARN ITS BODY GEOMETRY FOR OBSTACLE AVOIDANCE

This first experiment shows how an obstacle avoidance strategy can emerge as the effect of the interactions between the Khepera* robot (Fig. 1) and its environment, without the need of a geometrical model of the robot's body. Braitenberg[2] has already proposed an extremely simple control system for obstacle avoidance, but it is hard-wired, whereas we are concerned with learning of that behaviour. The whole robot's "brain" is based on unsupervised neural maps.[5] Our choice for neural networks is motivated by their generalization capacities, as well as their resistance to noise and contradictions. Such features are very precious for controlling a real mobile robot which is to operate in real-world situations. Furthermore, as they are directly inspired from animals whose behaviour is controlled by neurons, a global solution to the behaviour emergence problem can be expected to come from such tools.

Kohonen maps have been used for localization of the robot in its environment[14] but learning in such topological maps requires a separation between the learning and utilization phase. While a similar process may happen in biological systems such as building of somatotopic maps, it cannot be used for higher level applications where the environment is changing: when a new learning example is presented to a Kohonen map, all learning has to be done over again. It is thus impossible to make a gradual learning and to put decisions into question in a systematic way. Obviously, most learning in living systems does not start from scratch when a new item is encountered, which leads us to reject that algorithm and all the other algorithms based on statistical computations over a large set of data. Therefore, it is necessary to design a network that can learn incrementally, without prior knowledge of the training set. As the environment is subject to change, there should be no difference between learning and utilization. However, the information provided by topological maps is precious, especially when implemented in real systems: their main feature is that two close stimuli should produce close activity patterns in the map (Fig. 2); and when real sensors are used, noise should not perturb the consequent activities of the network. Close situations should be coded on nearby neurons.[10,11] This justifies the use of a new kind of topological map which is not fraught with the difficulties mentioned above.

Coding a special situation on certain neurons of a topological map does not allow the robot to adopt a particular behaviour. In this experiment, the aim is to learn how not to collide in walls. Therefore, we need a learning procedure that allows the robot to choose from all sensorial situations one movement that does not cause it to collide in a wall. We use for that level a winner takes all (WTA) and hebbian learning modulated by a pain/pleasure signal. At this stage, the whole process on the robot is discrete. Weights are initialized with random values. Then, at each time step, the following processes take place in order:

1. Sensorial information is collected.
2. A decision is made, a movement is performed.
3. A pain signal is computed and possibly perceived by the network.
4. Synaptic weights are adapted.

Movements are chosen among three possibilities: turn left, turn right, go straight ahead, each of them from a fixed quantity. The global architecture of our neural system is depicted in Fig. 3. Data from the infrared sensors are pre-processed in the "sensors input" box. It provides binary information about the position of the maximum of intensity variation between adjacent sensors and about the open directions and the wall's proximity. Two neural groups are used: one topological map that codes sensorial input and a WTA group that decides what movement to make. The topological map is a probabilistic topological map (PTM).[9]

The WTA group receives input from a reflexes box and from all cells of the topological map. The reflex of going forward is represented in a neural fashion: three neurons are used, with activity 0 (for the turning

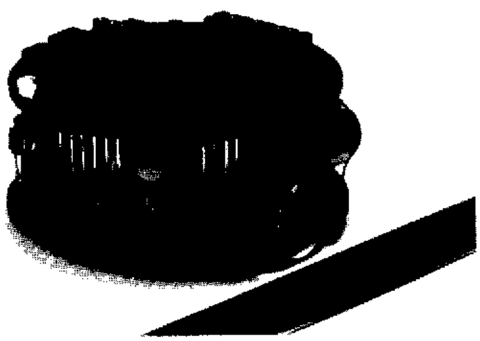

Fig. 1. The Khepera robot developed at LAMI.[13]

*Khepera is a robot designed by the K-Team at LAMI-EPFL.[13]

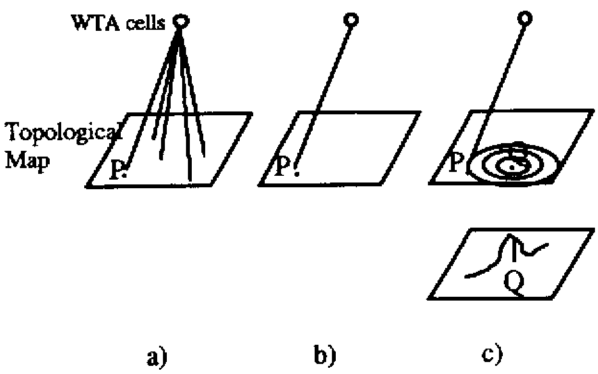

Fig. 2. Usage of the topological map. (a) Each neuron on the winner takes all (WTA) group receives afferences from all cells on the map. (b) P is the winner. Only this weight is modified by reinforcement learning. (c) When a new stimulus is presented to the map, Q is the winner. The activity in the WTA decreases with the map distance between Q and P.

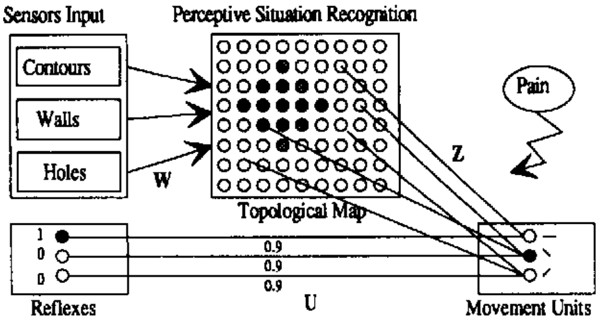

Fig. 3. The neural network architecture. The broken line associated with pain represents chemical transmitters released to enhance learning.

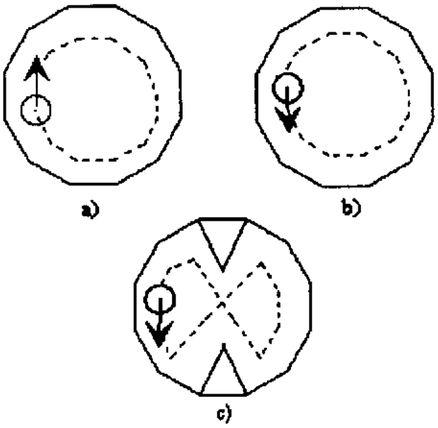

Fig. 4. Different trajectories of the robot, depending on the initial situation.

movements) or 1 (for the forward movement when there is no obstacle ahead). The activity they produce in the WTA is their own activity weighted by weight U. This weight is not modifiable, and the stimulus is called unconditional after Pavlov. On the contrary, links between the topological map and the WTA—called Z weights—must be adapted in functions of the pain signal.

This architecture is well adapted to the defined task: with an appropriate learning rule on the Z weights, it is possible to avoid making movements that cause pain when a similar situation has been met before. The one we use is inspired from Barto's reinforcement learning.[1]

In order to illustrate the convergence properties of the whole neural network, we launched the robot in an arena, where obstacles can be introduced or removed. This makes a good field to study how the robot learns to avoid collisions with obstacles. The rotation angle is set around 30°, and the forward step to about 1.5 cm, that is, about half the sensor's range. Figure 4 illustrates one experiment that was done with the robot. After a few steps, it finds itself with the wall on its left, and goes over the arena, keeping the wall on its left [Fig. 4(a)], and turning left before or after bumping on the wall. The robot has learned to associate each sensorial situation it has met with the "turning right" movement. If we then take the robot and turn it so that the wall is on its right [Fig. 4(b)], after a few steps it starts turning right and avoids collisions in the same way it did before. After these two experiments, the robot is placed in an arena loaded with new obstacles [Fig. 4(c)]. It then takes an eight-shaped trajectory, turning right when the wall is on its left, and left when it is on its right. This shows that it has learned the right associations for all the sensorial situations and that it has not forgotten anything. Thus, Khepera exhibits a kind of "intelligent" behaviour at this stage, as its ability of turning in the "good" direction was not decided in advance. It could indeed always turn in the same direction, say left, until it does not collide anymore, whatever the initial position towards the wall. This example shows it is possible to design a neural network that allows our robot to exhibit a behaviour which has not been programmed. Our neural architecture proves to be very effective:

1. No idealization of the sensors or of the outside world was needed. The primitives extracted from the infrared sensors are very robust to noise, and lend themselves extremely well to neural processing.
2. The topological map is organized while the behaviour is emerging. This happens very fast (around 100 steps), if it is to be compared with Kohonen maps[11] which generally need some 10,000 steps before organization appears.

Another striking feature of this architecture is that it is used exactly as such for totally different applications such as recognition of visual scenes[8] and landmark-based navigation. This leads us to think that this type of architecture might serve as a building block for more integrated tasks.

3. ACTING ON OBJECTS AS A MEANS TO CHANGE THEIR PERCEPTION

Here, we illustrate with one example that acting on the environment simplifies the reasoning needed by a robot and allows clustering of scattered objects. This experience also provides information about how simple robots can interact with their environment and cooperate to perform a task in an emergent fashion. We have used four Khepera robots (Fig. 1) to simulate a clustering task by ants.[4] The robots must collect pieces of wood and bring them together into clusters. They are programmed to move randomly and to take an object if they find one and to deposit it if they find another object. With this algorithm, the robots should bring objects and build little heaps made of only two objects, then destroy the heaps to build other ones, and so on. Nevertheless, the robots do build larger heaps [Fig. 5(a)]. This is due to an interpretation error! When a robot puts together two objects they can be perceived as a wall and the robot then avoids them; so the probabilities of picking up or putting down an object depend on the robot orientation with regard to the heap. For the same reason, the robots tend to make the clusters long rather than circular.

In this implementation, where several Kheperas are put on the playground simultaneously, it is not

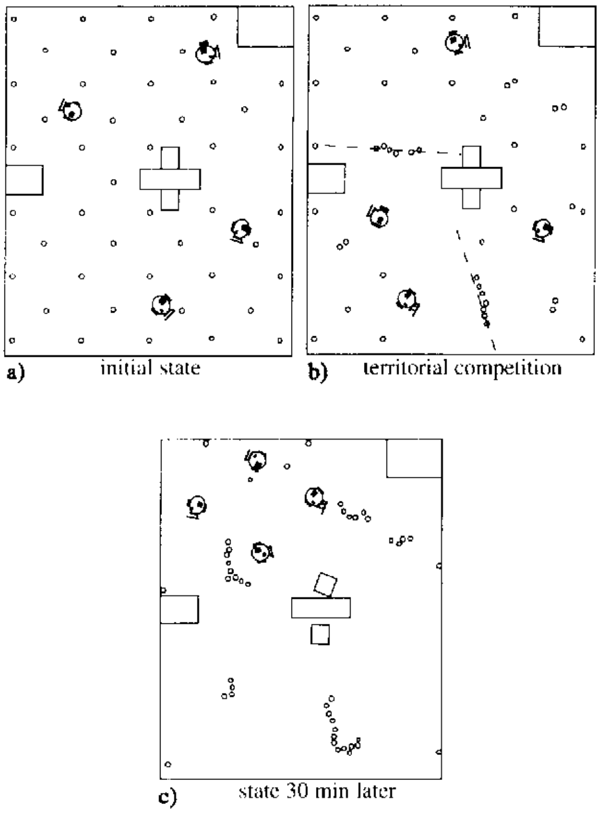

Fig. 5. A collecting task for four robots. (a) The initial situation. (b) During process: linear heaps are being built. This corresponds to a competition between robots for given zones. (c) The situation after 30 min.

possible to control the probabilities as in simulations; they are imposed by the physics of the problem. When a heap is met under a certain angle, then it can be taken for an obstacle and the object carried is not put down, which illustrates clearly how action modifies perception. Those conditions lead the heaps to take a somewhat linear form, which in turn divides the space into "closed" zones in which certain robots specialize [Fig. 5(b)]. However, the position of the clusters cannot be known in advance. The kind of equilibrium reached by this interaction is essentially dynamic: the size of the clusters is never fixed as there is always a chance that a robot seeing the heap from the right angle takes away an object. However, there is a statistical equilibrium [Fig. 5(c)].

The individual behaviour described above can be implemented by the neural network depicted in Fig. 6:

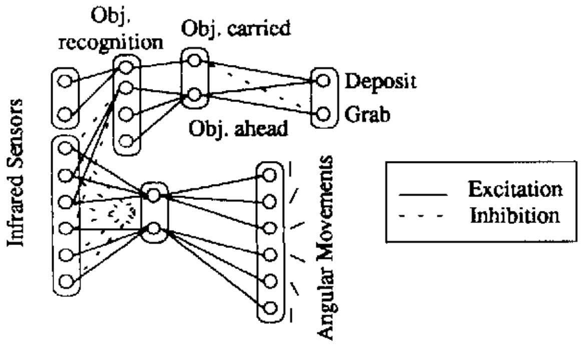

Fig. 6. A neural network for object collecting behaviour.

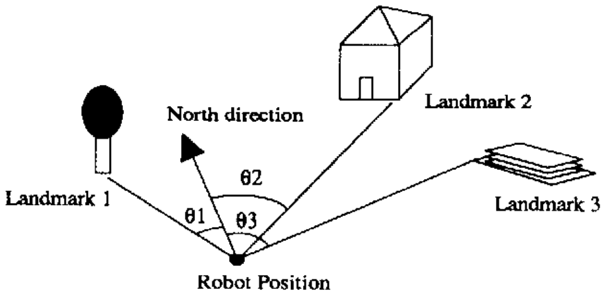

Fig. 7. Example of a landmarks configuration that the robot can use in a localization task.

infrared sensor values are an input to the obstacle avoidance mechanism and to the object recognition group. This last group is necessary in order to tell the difference between an obstacle and an object. Its synaptic links to the sensors are positive from two contiguous sensors and negative from the next on each side. Thus, more than two saturated contiguous sensors are not recognized as a grabbable object.

An interesting action occurs when one Khepera takes one of its mates for an object: it leaves the object it carries, and the second Khepera grabs it. This cooperation of two robots can be seen as a form of elementary communication which emerges through interactions with the environment.

On another level, when the robots move objects they create frontiers and isolated domains in which the other robots cannot go easily. The robots then have no choice but to work in a particular area. When the objects in an area are all joined together the robot goes away to find another zone to collect objects. Here, moving objects has become a tool to specialize the robot's work in a given area. Therefore, there is competition between the robots to share the clustering work although cooperation between the robots can also appear, for instance when a robot gives an object to another robot. This competition/cooperation mechanism is to be compared with the same paradigm used in neural networks.

4. ACTING AS A MEANS TO SIMPLIFY GOAL RETRIEVAL

Now, we can show how simple neural mechanisms can explain supposed complex cognitives capabilities. We simulate a mobile robot that can learn to retrieve a given position without any global knowledge about how to navigate. We suppose the existence of a visual system that is able to recognize landmarks in the environment and that can measure the angle between a landmark and an absolute direction like the north or the sun or any special landmark distant enough (see next paragraph for further details). A simple neuron called a "place cell" can then learn a particular location[16] and react according to the proximity of the robot to this stored location. Such a neuron seems to have been found in the brain hippocampus by neurobiologists. It is therefore a good candidate for designing a robot with navigation capabilities but

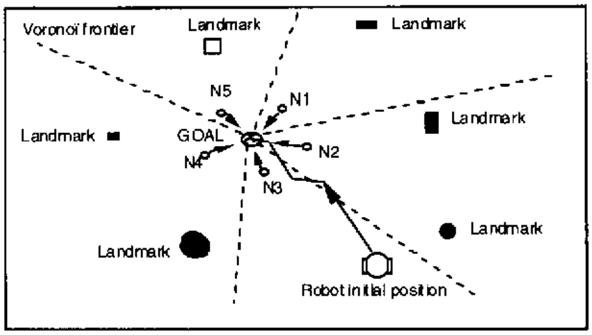

Fig. 8. Local exploration around the target represented by the large black circle. The agent records at certain points (represented by small circles) the positions relative to the landmarks (represented by squares) and the direction to the target.

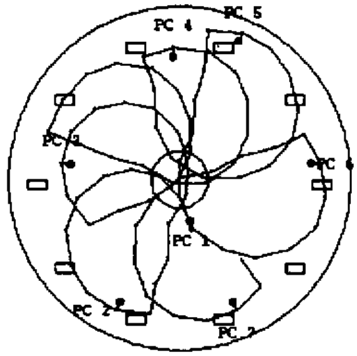

Fig. 9. Local exploration around the target to learn how to retrieve it. We assume that the target can be seen from any point inside the inner circle. The robot records at certain points (represented by black dots) their positions relative to the landmarks (represented by rectangles) and the direction to the target. See text for explanations about the trajectory. The numbers correspond to the place-field number in its neuron group.

without any *a priori* Cartesian map of its environment.[16]

At the beginning, we suppose the robot moves randomly. When it finds something interesting such as "food", it moves around it and learns that from a particular location it can go to the target by performing a movement in a given direction. Later, when the robot wants to find "food", it considers the information of the place cells associated to the food and moves one step in the direction associated with the most activated place cell (competitive mechanism). Thus each time the distance to the target is reduced (Fig. 8) and the robot inevitably returns to the learned position of the food.

The learning phase is more complex because it is an unsupervised and on-line process. When the robot "eats food", it triggers a reflex which allows it somehow to circle around the food at a certain distance, in order to visit evenly placed locations around it. At each of these well-chosen locations, a place field learns the relative position of the robot to the landmarks, and the direction heading towards the target (Fig. 9).

We propose here a complete neural network for landmark-based navigation (Fig. 11). Connecting it to a vision network based on the same idea, as proposed in Refs 7 and 8, would yield a "brain" for a robot, allowing both object recognition and target retrieval. We focus mainly on designing the simplest architecture for the desired behaviour, and on the small number of building blocks used. The main computing part is a PerAc block. It involves three neuron groups, all of the PTM type. Two one-dimensional PTMs are used to represent movement directions and one two-dimensional for localization, i.e. place cells. There is an internal representation of the world expressed in a referential independent to the robot's orientation with respect to its surroundings. As a consequence, two groups are used for movements, because one must correspond to movement directions with respect to an absolute direction and be associated with localization, while the other corresponds to the movement actually performed by the robot, which means that it takes the robot's orientation into account. When a movement direction is selected, the robot makes one step of given length in that direction. The inputs to this network are the north direction, and the food and landmark positions in the robot's visual space. We assume that a compass is available and provides the information for the former, while we take for the latter a processing of the vision network output. We also assume that the immediate visual angle is limited, just as for humans. Therefore, "food" is perceived only when it is located in a given orientation ahead of the robot. The same goes for the landmarks, but we assume that when a position must be recorded, the robot rotates in order to see in all directions. This supposes that when exploring a scene, the robot can make ocular saccades and move its head as well, thus spanning the whole surrounding space.

The functioning of the neural network is easier to understand when starting from the end, that is the one-dimensional PTM groups corresponding to the movements RM (robot movement) and ERM (effective robot movement). We used two different groups, because the "exploration" reflex must activate a "turn left by a certain angle" from the actual angular position of the robot. On the other hand, learning of place cells and of associations must be done in a fashion which is independent of that position, thus producing an internal representation of the robot's world. Indeed a place cell represents a position in space, not an orientation of the robot. Therefore, its activation should depend only on the position of the robot, which is obtained by using a shifting mechanism similar to the one described about vision. In the same manner, the first WTA should record directions of movement independent of the robot's direction.

When food is in sight, a neuron corresponding to its angular position relative to the robot's facing position is activated in the UMI group. The shifting mechanism activates a neuron in the invariant food position by adding an angle corresponding to the angle between the robot and the north. This invariant position activates a neuron in RM which corresponds to movements relative to the north. If there is pleasure at that moment, a place cell learns the invariant landmark's position, and the association with the movement in RM. The inverse shifting mechanism is applied

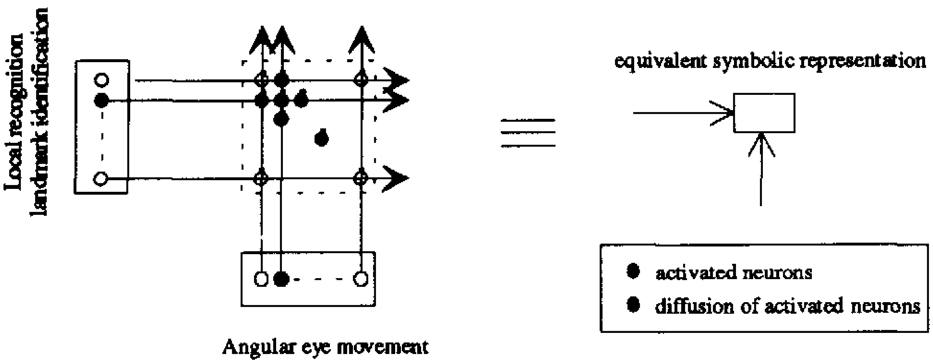

Fig. 10. Recombination of visual and motor flow as an input to the place fields cells.

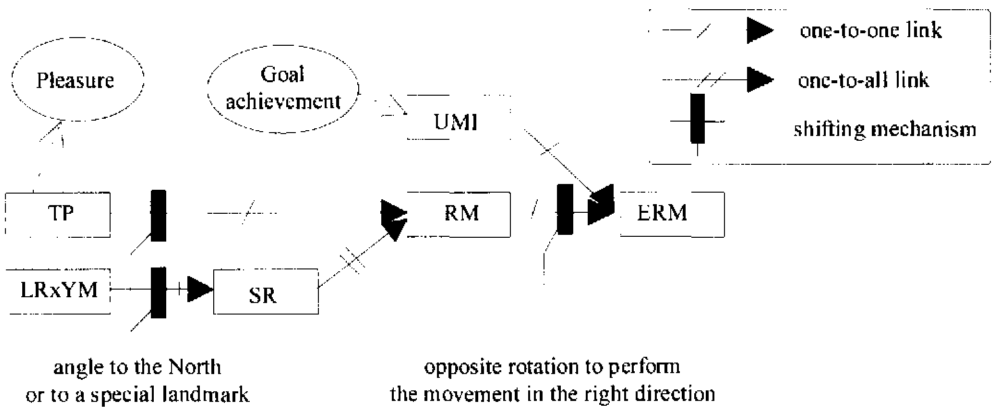

Fig. 11. The navigation neural network. SR is the scene recognition group. We take a PTM as a model for it. Its input is the LR × YM group which corresponds to the landmarks recognition (LR) associated with the eye movement (YM). The robot movement (RM) group is a one-dimensional PTM. It codes the chosen direction with respect to the position of the north. When the target (food) is visible [target proposal (TP) group], the chosen direction in RM corresponds to the food position, because of high-valued one-to-one links between the TP and RM groups. The ERM group is also a one-dimensional PTM and it corresponds to the effective robot movement (ERM) in the environment. It gets one-to-one afferences both from RM and from the unconditional motor input (UMI) group. This latter group is active for a given time span starting after reaching the food. Because of a high-valued link to ERM, the reflex has a priority when active, causing the robot to turn in a given direction, thus giving rise to ellipsoidal trajectories. The black rectangles represent a shifting mechanism used either to provide an invariant representation of the input or to transform invariant representations into actual ones. Pleasure is emitted when the food is in sight. It functions as a chemical substance emitter, and enables learning throughout the whole network.

to the output of that group, by subtracting the same angle. This activates the neuron in ERM which corresponds to the actual movement to be performed by the robot. If the exploration reflex is active, then its corresponding neuron wins over all the others because the link between reflex and ERM is higher than all the others. The unconditional movement (UMI) is triggered when the robot reaches the food and remains active for a certain amount of time. The provoked trajectories after reaching food thus take an ellipsoidal shape, which ends after a while. As soon as food is in sight (given a limited visual angle) the position of the landmark is recorded. This supposes that when pleasure is active, the robot moves its "head" in order to see landmarks in all possible directions.

In Fig. 12, we represented 200 simulated trajectories for different starting positions. Using a probabilistic activation law in the ERM group allows a smoothing mechanism, as it computes an average direction between the most active motor neurons. By using an inhibition mechanism between the place cells group and ERM, it is also possible to get the robot to learn to avoid a given zone. The probabilistic competition mechanism then ensures that the robot takes a trajectory close to the forbidden zone.

5. CONCLUSION

Throughout this paper, we have insisted on the importance of active perception. We have shown through all the examples proposed that using action simplifies the interpretation of perception: a choice is made at each time step, which conditions entirely the future of the robot. It should also be noted that the introduction of noise in the system output of the system generates diversity and a "personalization" of the behaviours. The greatest advantage of this type of approach is that it makes cognition sequential, thereby avoiding the possible large duplications and relaxation mechanisms needed by massively parallel systems such as the connectionist systems proposed by Feldman or by the PDP group (McClelland and Rumelhart). Indeed, if all recognition tasks were performed in

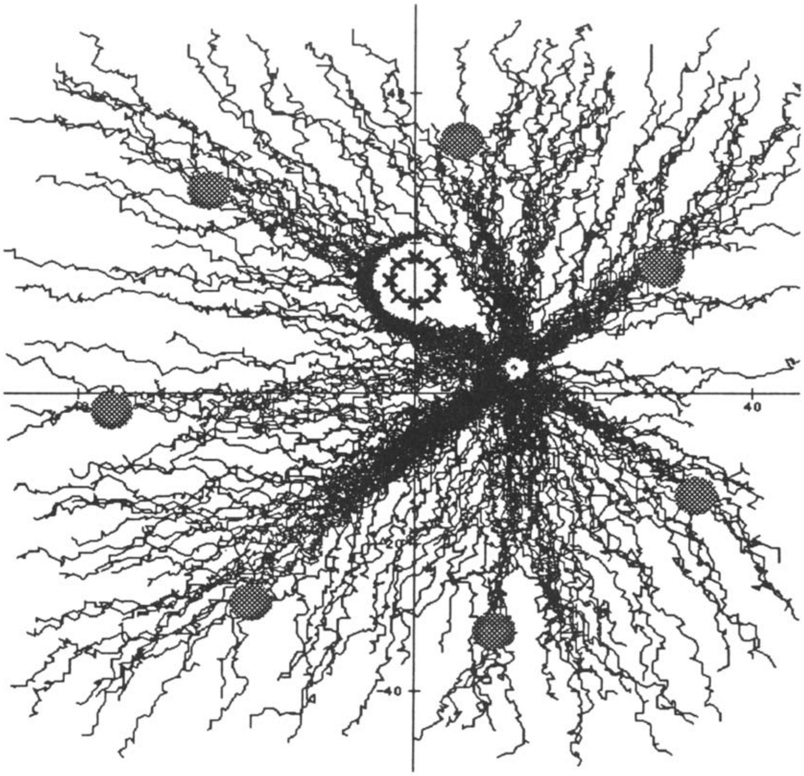

Fig. 12. Two hundred simulated trajectories using the neural network of Fig. 11, with a probabilistic competition mechanism and inhibiting links. Landmarks are represented by large grey dots. The forbidden zone is surrounded by crosses representing the inhibiting place cells. The target is at the intersection of all trajectories, at the middle of the small empty area.

parallel without a time element, one should duplicate the recognizing element as many times as a recognizable object is present. If the processing is sequential, one element is needed for each type of objects, and a procedure to perform recognition as many times (sequentially) as there are prototypes of this object in the image.

We have put forward a basic building block, the PerAc block, which provides a systematic way of performing a fusion of motor and perceptive information. Due to the above remarks, we are keen to believe that this unit is the minimum required for a relevant approach to cognitive problems. From the cognitive point of view, the interest of this processing architectures is that the sensory-motor categories, i.e. the association of sensory situations with proper actions, are formed from the point of view of the agent, and not according to some criteria predefined by the operator.

Future work will be concerned with finding ways to optimize the architecture parameters, and extending networks to more complex tasks, always relying on the constructed level to obtain the next. Particular attention will be paid to what must be stored in the robot "genotype" and how to improve its cheap limbic systems, which is the major element for controlling the robot motivations and allowing learning and adaptation capabilities; so a robot does not need to be so smart that we need to design all its features case by case. To take into account dynamics of the robot and to use learning mechanisms seem to be good ways of avoiding the world model trap of classical artificial intelligence.

REFERENCES

1. Barto, A. G., Sutton, R. S., Anderson, C. W.: Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Trans. Systems, Man, Cybern.* **SMC-13**: 834–846, 1983.
2. Braitenberg, V.: *Vehicles: Experiments in Synthetic Psychology*. Cambridge, MIT Press, 1986.
3. Brooks, R. A.: A robust layered control system for a mobile robot. *IEEE J. Robotics Automation* **RA-2**: 14–23, 1986.
4. Deneubourg, J. L., Goss, N., Franks, N., Sendova-Franks, A., Detrain, C., Chrétien, L.: The dynamics of collective sorting: robot-like ants and ant-like robots. In *European Conference on Artificial Life*. Cambridge, MIT Press. 1991.
5. Edelman, G.: *Neural Darwinism: The Theory of Neuronal Group Selection*. New York, Basic Books. 1987.
6. Edelman, S., Weinshall, D.: A self-organizing multiple view of 3D objects. *Biol. Cybern.* **64**: 209–219, 1991.
7. Gaussier, P.: Doctoral thesis, Paris XI-Orsay, 1992.
8. Gaussier, P., Cocquerez, J.-P.: Neural networks for complex scene recognition: simulation of a visual system with several cortical areas. In *Proceedings of IJCNN*, Baltimore. 1991, Vol. 3, pp. 233–259.
9. Gaussier, P., Zrehen, S.: The probabilistic topological map (PTM): a self organizing and fast learning neural map that preserves topology. Submitted to *Neural Networks*

10. Hubel, D., Wiesel, T.: Functional architecture of macaque monkey visual cortex. *Proc. R. Soc.* **198**: 1–59, 1977.
11. Kohonen, T.: Self-organized formation of topologically correct feature maps. *Biol. Cybern.* **43**: 59–69, 1982.
12. Maturana, Varela: *The Tree of Knowledge.* Boston, Shambhala. 1987.
13. Mondada, F., Franzi, E., Ienne, P.: Mobile robot miniaturization: a tool for investigation in control algorithms. In *Proceedings of the Third International Symposium on Experimental Robotics.* Kyoto. In press.
14. Nehmzov, U., Smithers, T.: Using motor actions for location recognition. In *Proceedings of the First European Conference on Artificial Life*, Paris, Varela, F. J., Bourgine, P. (Eds). Cambridge. MIT Press. 1991, pp. 96–104.
15. Stewart, J.: Life = Cognition: The epistemological and ontological significance of artificial life. In *Proceedings of SAB 91*, Paris, Bourgine, P. Varela, F. (Eds). Cambridge. MIT Press. 1991, pp. 475–483.
16. Zipser, D.: A computational model of hippocampal place fields. *Behav. Neurosci.* **99**: 1006–1018, 1985.