Modeling and Simulation of Elementary Robot Behaviors using Associative Memories

Claude F. Touzet
Adaptive and Integrative Neurobiology, UMR 6149, University of Provence / CNRS
Centre St Charles - Pôle 3C - Case B. 3, Place Victor Hugo, F - 13331 Marseille Cedex 03, France
(Part of this work was conducted during a previous position at Center for Engineering Science Advanced Research, Computer Science and Mathematics Division, Oak Ridge National Laboratory, TN, USA)
Claude.Touzet@up.univ-mrs.fr

Abstract: *Today, there are several drawbacks that impede the necessary and much needed use of robot learning techniques in real applications. First, the time needed to achieve the synthesis of any behavior is prohibitive. Second, the robot behavior during the learning phase is – by definition – bad, it may even be dangerous. Third, except within the lazy learning approach, a new behavior implies a new learning phase. We propose in this paper to use associative memories (self-organizing maps) to encode the non explicit model of the robot-world interaction sampled by the lazy memory, and then generate a robot behavior by means of situations to be achieved, i.e., points on the self-organizing maps. Any behavior can instantaneously be synthesized by the definition of a goal situation. Its performance will be minimal (not necessarily bad) and will improve by the mere repetition of the behavior.*
Keywords: *Robot learning, Kohonen map, self-organizing map, autonomous robotics, associative memory programming, obstacle avoidance.*

1. Introduction

1.1 The Future of Robotics

Learning is a necessary component of robotics for reasons as serious as the time and money required to write ad-hoc behaviors, or simply because an accurate-enough model of the environment may be unavailable, as is the case of space exploration, sub-marine exploration, or nuclear powerplant assessment (after an accident).

1.2 Expensive vs. Cheap learning

The two most widely used robot learning paradigms are supervised learning and reinforcement learning. Supervised learning (Le Cun, 1985 ; Rumelhart, Hinton & Williams, 1986) requires the operator to define a set of representative examples of situation-action pairs (i.e., the learning base). On the other hand, reinforcement learning (for a review see: Kaelbling, Littman & Moore, 1996; Sutton & Barto, 1998) generates the learning base through a combination of an exploration and a reinforcement function. In the latter case, the operator is only asked to define a measure of the robot behavior performance. Despite the efforts to come up with a reinforcement function design process (Santos & Touzet, 1999a, 1999b), a lot of time is spent in trial and error. Moreover, a reinforcement function has to be defined for each desired behavior, which means that – even if the reinforcement function is perfect – a new learning base must be build for every single behavior.

1.3 Faster learning

Lazy learning (Aha, 1997) reduces the time required to build the learning base. In an initial and unique sampling of the robot-environment relation, lazy learning builds a non-explicit model of the situation-action relation. Coupled to a reinforcement learning technique, such as Q-learning, lazy learning allows a great reduction of the time necessary for learning. The learning iterations are done in simulation using the non-explicit model, much faster than the actual time needed by a robot to performed the required number of actions. Lazy Q-learning (Sheppard & Salzberg, 1997) is becoming a paradigm of choice for robot learning, allowing almost instantaneous behavior synthesis - distributed lazy Q-learning techniques have already been proposed in the multi-agent context (Darrell, 1997 ; Touzet, 2005).

1.4 Limitations of lazy Q-learning

Lazy Q-learning application development is still time consuming. During development, many different reinforcement function expressions appear valid, and only experimentation is able to verify the quality of the synthesized behaviors. Research efforts in reinforcement function design only help the learning to converge, but

there is no warranty that it will converge towards the desired behavior. This is due to the highly indirect way the behavior is synthesized. A behavior is seen as a mapping between situation and action, and the learning is a function approximation method that uses generalization over a subset of high utility situation-action pairs, gathered during exploration. The utility, initially a qualitative information, is transformed into quantitative values by the training rule. Being able to imagine a priori the behavior that will emerge from such complex process is very difficult, and progress in exploration, training rules, and generalization are not going to offer a definitive solution.

1.5 Goal-seeking behaviors

On the other hand, if the desired behavior is expressed not as a mapping between situations and actions, but as a situation to achieve (a goal to seek) then there is a direct relation (not necessarily a bijection) between the goal to be achieved and the representation of this goal in the operator's mind. Until today, goal-seeking methods in autonomous robotics have provoked little interest. They are mostly related to mapping applications, such as the go-to-the-nest application (Sehad & Touzet, 1995). They associate a utility value with each situation-action pair encountered during the learning phase, which is later used as an indication of which action to choose at a given location. If the position of the goal changes, then the learning must be started again. Applications to other domains, such as collision avoidance behavior (Touzet, 2003), encounter the same limitation. Only one behavior is learned and the learned behavior cannot accommodate changes in the shape or size of the obstacle.

1.6 Summary of the paper

In the following section 2, we present our method used to immediately generate a behavior by locating intermediate situations to reach on the self-organizing map. Experiments synthesizing an obstacle avoidance behavior for the Nomad 200 mobile robot are presented in section 3. Section 4 presents the related works and, finally, we conclude and offer a few ideas that extend and complete the learning method described in this paper.

2. Our model

2.1 Robot behavior's definition

We propose defining a behavior by means of a desirable goal for the robot to achieve. Therefore, the robot must be able to perceive such an achievement. The goal is then a robot sensory situation that is desirable. For example, it can be a perceived situation completely free of obstacles in the case of an obstacle avoidance behavior.

2.2 Straight-forward limited implementation

A behavior is a mapping between situations and actions. Dynamically, a behavior can be represented as a sequence of points in the situation-action space, each point belonging to the mapping. A sequence of points defines a trajectory, or a path. Generating the desired behavior is then producing the sequence of actions that will take the robot from its initial situation to the goal situation (assuming that the goal can not be achieved with only one action). The problem is that - due to the high dimensionality (usually much larger than 3) of the situation-action space – the number of situations for which an action has been tried is too small. Despite the use of lazy learning, the ratio of situation-action pairs over the search space size is extremely small. Therefore, for each current situation, there are extremely few similar ones. Most of the time, not enough situation-action pairs have been sampled to allow a reasonable selection - the number of points needed grows exponentially with the number of dimensions of the situation-action space. In conclusion, using lazy learning, there is no guarantee that the selected course of action will lead the robot to the goal.

2.3 Associative memories

Associative memories are memories accessible using part of their content. The self-organizing map (SOM) (Kohonen, 1987, 2001) is one of such associative memories. It is a clustering technique that adds a neighborhood property between the clusters (this unique properties explains its thousands of applications listed in (SOM-database, 2001 ; Oja & Kaski, 1999)). Neurons (clusters) can be neighbors, or not.

The number of neighbors per neuron specifies the dimensionality of the map. For example, 4 neighbors correspond to a 2-D map. Unlike many clustering techniques, the number of neurons/clusters is fixed and predefined (at least in standard SOM versions). The positions of the clusters in the input space are defined by the input distribution and the neighbor's positions (Fig. 1a and 1b).

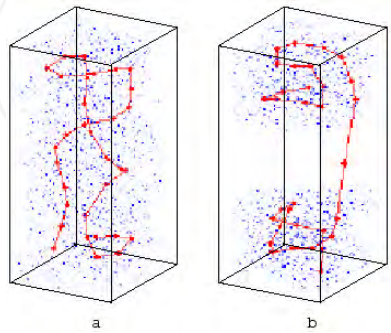

Fig. 1. The 32-neuron SOM maps the 3-D space into the 1-D SOM space (neighborhood of 2). Each neuron is represented by a circle and the lines drawn between neurons show the neighborhood. (a) The inputs are a set of uniformly distributed points (dots). Each neuron represents 1/32 (on average) of the input space (b) The inputs belong to 2 sets: Neurons code preferably these 2 regions, but the continuity property forces a few neurons to code space regions devoid of input points.

The size of the clusters is directly related to the distribution. Each neuron corresponds to (approximately) the same number of input points. Therefore, regions where the input data are scarce will generate neurons with larger fields of attraction than regions of greater density.

2.4 Associative memories implementation

Using two self-organizing maps, we are able to provide the flexibility that is currently missing in goal seeking robot learning methods:

- The first self-organizing map builds a continuous representation of the situation space (fig. 2). This representation, for example a 2-D map, can be used to find a path – a set of intermediate situations – towards a goal situation, wherever it is.
- The second self-organizing map is used to generate the action that will change the sensory inputs from the current perceived situation to the computed intermediate situation.

The ability of the maps to represent the actual distribution of inputs allows for a greater discreteness in the areas of interest, i.e., the behavior performance will improve with repetition. Learning occurs by the mere realization of the behavior.

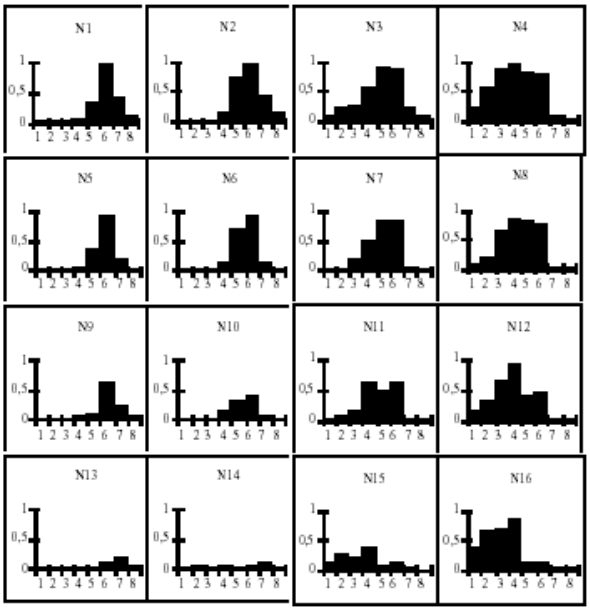

Fig. 2. 16 situations coded by the 16 neurons of the SOM after 100 iterations of learning. The behavior of the robot during this sampling phase is random walk. Situations are represented by a vector of 8 components. A value of 1.0 (maximum) is associated to a distance to the obstacle shorter than 2 cm. A value of 0.0 is associated to an absence of obstacle (or an obstacle at least 5 cm away). For example, neuron N1 codes for an obstacle 2 cm away on the right side of the robot ; N2 codes for larger obstacle in the same direction. Similar situations are coded by neighbor neurons.

2.5 Algorithm: associative memory programming (fig. 3)

1. The neighborhood conservation property allows the definition of a metric on the input space. Let us say that the first SOM maps the situation space, then it becomes possible to choose among the neurons a neighbor neuron/situation closer to the goal, which will be the intermediate situation to achieve.
2. Having found the intermediate achievable situation, we can now obtain the action that will move the robot into that desired intermediate situation using a second SOM.
3. This procedure must be repeated until the robot is in the desired goal situation.

3. Simulations

Experiments (fig. 4 & 5) have been conducted in synthesizing an obstacle avoidance behavior for the Nomad 200 mobile robot (fig. 6). The goal is defined as a perceived free-of-obstacle sensory situation. The resulting behaviors do not bump into obstacles and are of a similar quality as the best results achieved with reinforcement learning (Touzet 1997, 2003), but need no learning iterations. The associative memories (16 neurons per SOM) use the knowledge gathered by a random exploration of - only - 100 moves (compared to 200 minimum for Q-learning). And the same 100 iterations can be used to generate another behavior instantaneously, such as an obstacle avoidance that avoids by the right side, wall following or go-to-the-nest behaviors.

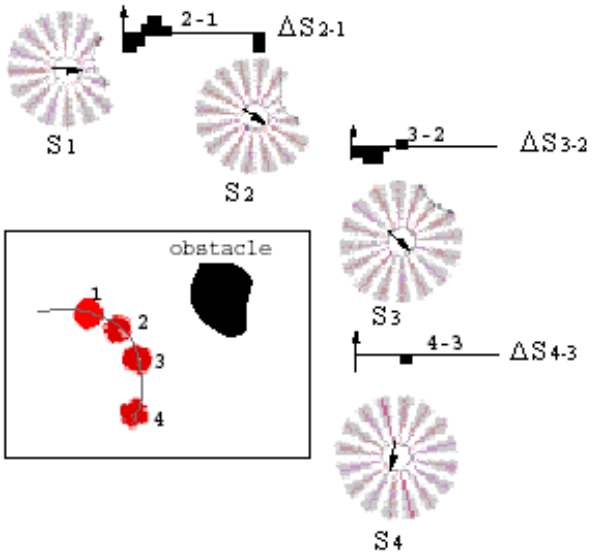

Fig. 4. Example of an obstacle avoidance sequence of situations. In the box (bottom left), the obstacle and successive positions of the robot are shown. The values of the 16-sensor sonar ring of the Nomad 200 are displayed. The histograms show the sensory differences between two neighboring situations (St – St-1). These differences are the inputs for probing the second SOM to retrieve the actions.

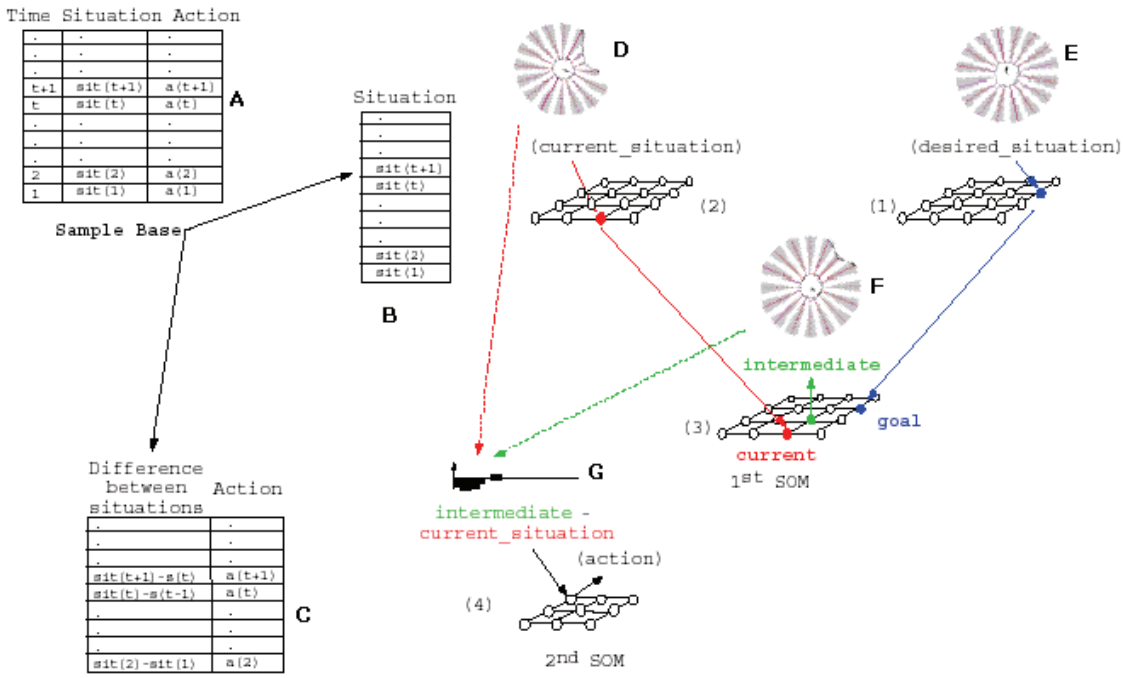

Fig.3. The lazy memory (top left) is used to build the 1st SOM that maps the situation space, and the 2nd SOM that maps the difference between sensory inputs vs. the action. The samples of the two learning bases have been obtained using a random action selection policy for the robot. The behavior is generated through the following process: (1) Find the neuron corresponding to the desired goal situation. (2) Find the neuron corresponding to the current situation. (3) Find a neighbor neuron (of the current situation neuron) closer to the "goal" neuron which will represent the intermediate situation to achieve. (4) Probe the 2nd SOM with the sensory variation between the intermediate situation and the current situation to get the action that must be carried out.
(A) is the sample base, (B) is the learning base of the 1st SOM, (C) is the learning base of the 2nd SOM, (D) is the current situation with an obstacle on the left, (E) is the goal situation with no obstacle in view, (F) is the intermediate situation to achieve, (G) is the difference between situations (F – D).

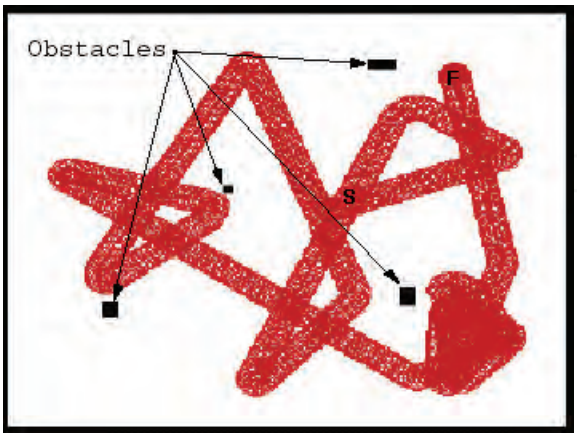

Fig. 5. Obstacle avoidance behavior generated by a Nomad 200 (mobile robot) using its sonar sensors. The position of the robot is indicated by a circle. The robot starts in the center of the area (S: start position) and moves towards the right (F: final position). It avoids obstacles such as wall from a greater distance than the smaller obstacles – but it does not bump into any obstacle. The SOMs use the knowledge gathered by a random exploration of – only – 100 moves. The goal is defined as a perceived free-of-obstacle sensory situation. Both SOMs use 16 neurons and a neighborhood of 4.

Fig. 6. The Nomad 200 is a mobile robot equipped with a ring of 16 sonar sensors.

4. Related works

Coiton et al. (Coiton 1991) propose a neural network model for a sensory-motor system composed of a

sensory layer (a SOM) and a motor layer. The objective is the generation of goal directed movements using a real robot arm. They show that their model is actually able to control the displacements of the robot arm. The input situations are three Cartesian coordinates and the outputs are three joint angles. The neural model learns the mapping between both sets of coordinates – only one behavior is possible. It is important to note that in our approach any behavior can be generated.

Smith (Smith 2002) in a tentative to achieve situation/action mapping in reinforcement learning, proposes to use two SOMs, one to map the situation space, the second to map the action space. Q-learning is used to compute the utility of each (situation, action) pair. As in the previously cited work by Coiton et al., his experiments involved the learning of a mapping from goal space to arm space, with an arm having as much as 20-dimensions. The reinforcement signal used is related to the distance (but not direction) to optimality. Despite the fact that two SOMs are used, only a single behavior can be learned, and it must be learned from scratch.

Laurence, Trappenberg and Fine (Laurence et al., 2005), following an initial idea from Lisman (Lisman, 1999) propose to use a pair of connected recurrent associative networks to generate simple temporal sequences of patterns. The various elements of a sequence are recalled one after the other. As soon as one of the associative memories converges towards a learned element, this element pattern is used as input for the second associative memory to force convergence towards the following element of the sequence, and so on.

5. Conclusion and future research

5.1 Immediate synthesis of elementary behaviors

Using a set of two cooperating self-organizing maps, we have been able to demonstrate that any behavior – in fact any relation between situations and actions – can be generated. This solves the problem stated in the introduction: immediate behavior synthesis with goal seeking methods. Many different behaviors can be obtained using the same SOMs, by simply defining various goal situations, such as wall following or following another robot. The defined goals are independent of the robot geometry or actuators. The amount of supervision required by the human operator is minimal compared to other approaches such as supervised learning or reinforcement learning.

5.2 Chaining elementary behaviors

More than one sensory modality can participate in the expression of a behavior (e.g., color, odor, space, landmark, etc.). In our framework, each sensory modality will be taken care of by a couple of SOMs. These maps, carrying diverse representations, must be put together from time to time (at least to generate intelligent behaviors). As stated by Gallistel (Gallistel 1990), space and time are two predominant aspects of reality and must therefore be part of any stored record (i.e., situation-action pair). These two coordinates link the separate records within the same SOM (if the neighborhood property is not sufficient) and between SOMs. This linking between records is what could enable complex behaviors to occur. Note that the linking of records is accomplished a posteriori, at the time of the retrieval and that there is no implication of the retrieval process in the memorization process. A desired behavior is generated by positioning a goal in a map, whose spatial and/or temporal coordinates will then be used to retrieve other records in the same or different maps, generating automatically a sequence of sub-goals to achieve, i.e., the action plan. We are continuing our investigation in this direction.

5.3 Human behaviors

The universally known Penfield's Homunculus (fig. 7) (Penfield 1975) demonstrated that sensory inputs are mapped by the left cortex and motor outputs are mapped by the right cortex. Knowing that:

- a SOM is a plausible model for the neural organization of the cortex,
- our model is implemented by two SOMs, one for situations, one for difference in situations and the corresponding action,

we cannot avoid to hypothesize that some elementary behaviors in animals and humans may be generated by the same means (goal seeking behaviors). This could be in particular the case of small movements involved in pointing behaviors.

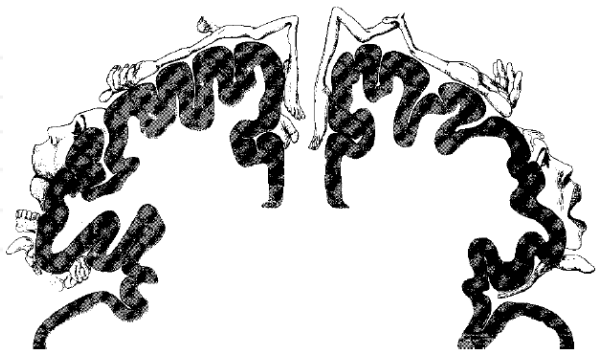

Fig. 7. The Penfield's Homunculus. Cortical surface areas are proportional to the number of sensors of the body part (and not proportional to the skin surface). Adjacent body parts are adjacent on the cortical map. The sensory cortex (left hemisphere) is devoted to sensory input representation ; the motor cortex (right hemisphere) is devoted to the control of action. Kohonen's SOM model (cf. §2) explains how such maps can self-organize.

6. References

Aha, D. (ed.). (1997). Lazy Learning, Kluwer Academic Pub.

Coiton, Y.; Gilhodes, J. C.; Velay, J. L. & Roll, J. P. (1991). A neural network model for the intersensory coordination involved in goal-directed movements. Biological Cybernetics, 66:167-176.

Darrell, T. (1997). Reinforcement Learning of Active Recognition Behaviors. Interval Research Technical Report 1997-045.

(http://people.csail.mit.edu/trevor/papers/1997-045/TR-1997-045.ps.gz ; verified July 29, 2005) - Portions of this paper previously appeared in Advances in Neural Information Processing Systems 8, (NIPS '95), pp. 858-864, MIT Press, and Intelligent Robotic Systems, M. Vidyasagar ed., pp. 73-80, Tata Press, 1998.

Gallistel, R. (1990). The Organization of Learning, MIT Press.

Kaelbling, L.; Littman, M. & Moore,. A. (1996). "Reinforcement Learning: A Survey," Journal of Artificial Intelligence Research 4:237-285.

Kohonen, T. (1987). Self-Organization and Associative Memory, Second Edition, Springer Series in Information Sciences, Vol. 8, Springer Verlag, Berlin.

Kohonen, T. (2001). Self-organising maps 3rd edition, Springer, Berlin.

Lawrence, M..; Trappenberg T. & Fine, A. (2005). A multi-modular associator network for simple temporal sequence learning and generation, Proceedings of the 13th European Symposium on Artificial Neural Networks (*ESANN'05*), April 2005, Bruges, Belgium.

Le Cun, Y. (1985). A learning scheme for asymmetric threshold networks, Proceedings of Cognitiva'95, Paris, France, 599-604.

Lisman, J.E. (1999). Relating hippocampal circuitry to function: Recall of memory sequences by reciprocal dentate-ca3 interactions. Neuron, 22(2):233–242.

Oja, E. & Kaski, S. (Eds.), (1999). Kohonen maps. Amsterdam: Elsevier.

Penfield, W. (1975). The Mystery of the Mind, Princeton University Press, (Toronto, Little, Brown & Co.).

Rumelhart, D. E.; Hinton, G. E. & Williams, R. J. (1986). Learning internal representations by error propagation In: Parallel Distributed Processing, Vol. 1, D. Rumelhart & J. Mc Clelland Eds. Cambridge, MIT Press, 318-362.

Santos, J. M. & Touzet, C. (1999a). Exploration Tuned Reinforcement Function. Neurocomputing, 28(1-3):93-105.

Santos, J. M. & Touzet, C. (1999b). Dynamic Update of the Reinforcement Function during Learning. Connection Science, Special issue on Adaptive Robots, Carme Torras guest editor, 11(3-4).

Sehad, S. & Touzet, C. (1995). Neural Reinforcement Path Planning for the Miniature Robot Khepera, Proceedings of the World Conference on Neural Networks (WCNN'95), Washington D.C., USA.

Sutton, R. & Barto, A. (1998). Reinforcement Learning, MIT Press Bradford Book.

Sheppard, J. W. & Salzberg, S. L. (1997). A Teaching Strategy for Memory-Based Control, In: Lazy Learning, D. Aha (Ed.), Kluwer Academic Publishers, 343-370.

Smith, A. J. (2002). Applications of the self-organising map to reinforcement learning. Neural Networks 15:1107–1124.

SOM-database (2001) www.cis.hut.fi/research/som-bibl/

Touzet, C. (1997). Neural Reinforcement Learning for Behaviour Synthesis. Special issue on Learning Robot: the New Wave, N. Sharkey (Guest Ed.), Robotics and Autonomous Systems, 22(3-4):251-281.

Touzet, C. (2000). Robot Awareness in Cooperative Mobile Robot Learning. Autonomous Robots, 8(1):87-97.

Touzet, C. (2003). Q-learning for robots, In: The Handbook of Brain Theory and Neural Networks (Second Edition), M. Arbib (Ed.), MIT Press, 934-937.

Touzet, C. (2004). Distributed Lazy Q-learning for Cooperative Mobile Robots. International Journal of Advanced Robotic Systems, 1(1):5-13.