Subsymbolic Action Planning for Mobile Robots: Do plans need to be precise?

John Pisokas and Ulrich Nehmzow
Department of Computer Science
University of Essex
Wivenhoe Park
Colchester CO4 3SQ
United Kingdom
e-mail: [jpisso,udfn]@essex.ac.uk

Abstract

In (Pisokas and Nehmzow, 2003) we presented a subsymbolic action-planning mechanism capable of autonomously determining task-achieving plans using previously *acquired* subsymbolic representations. However, that planning mechanism was unable to pursue plans towards unexplored areas of the robot's perceptual space. This paper addresses that inability of the subsymbolic action-planning mechanism. We present and compare two alternative approaches: the first exploits the topology-preserving nature of self-organising feature maps and the second incorporates radial basis function neural networks as predictors of perception. Experimental results and analyses for different tasks conclude the paper.

1. Introduction

This paper addresses the question of how a mobile robot can autonomously determine task-achieving sequences of actions without using symbolic, pre-supplied knowledge, but generating novel plans using previously *acquired* subsymbolic representations.

1.1 Related Work

Symbolic Planning for Robots

One possibility for achieving action-planning is to determine a world model manually, to specify permitted operators (actions) applicable to it, and to use standard Artificial Intelligence search techniques to find paths through the action space that will lead from an initial state to the goal state. STRIPS (Fikes and Nilsson, 1971, Nilsson, 1984) is an early example of this approach.

More recently, hybrid architectures were developed which employ symbolic planning techniques. They attempt to tackle problems related with symbol grounding (Harnad, 1990) by incorporating a reactive layer (Arkin, 1998). Well known examples are SOMASS (Malcolm and Smithers, 1990) and ATLANTIS (Gat, 1992).

However, there are substantial weaknesses to approaches involving symbolic representations:

- Because of the external identification of operators, only part of the behavioural space available to the robot may be exploited.
- Similarly, only part of the perceptual space available to the robot may be used, due to the difference of perception and experience of the world between robot and human designer.
- Finally, actual robot perceptions and actions may differ from those defined by the human designer (e.g. due to sensor noise), eventually leading to brittle and unreliable performance.

An alternative would be to let the robot acquire its own subsymbolic world representation, and to let it associate preconditions and postconditions with its behavioural repertoire autonomously.

Subsymbolic Planning for Robots

The bulk of research on subsymbolic planning to date has focused on simulated agents, rather than situated, embodied mobile robots. In (Baldassarre, 2001a, Baldassarre, 2001b), to name a recent example, simulated robots were able to select *one* optimal action, given a perceptual state and a goal. In contrast to such 'action selection' mechanisms, we present a mechanism for 'action planning' that allows the association of *multiple* actions with each perceptual state.

Regarding applications in robotics, (Zeller et al., 1996) for instance presents a solution to a path planning problem for a real robot arm. However, heuristics were introduced *a priori* (in the form of a gradient descent search) adding a *tacit knowledge*[1] component to the planner, making it task specific. A similar approach is taken by (Fomin et al., 1996) in a simulated path planning task. In contrast to this approach, we avoid heuristics or other task specific knowledge altogether in the planner.

Closest to our approach is the work of (Bugmann, 1997), who uses a robot capable of performing only two actions; 'turn left' and 'turn right'. In contrast we use a self-contained mobile robot capable of moving in two dimensional space. Furthermore, we employ a self-organising feature map (Kohonen, 1984) for clustering sensory perception, in a similar fashion as in (Nehmzow et al., 1991); as a result the effect of noisy data is minimised. (Bugmann, 1997) accounts for problems similar with those we faced in the work presented in (Pisokas and Nehmzow, 2003). However, we here present two new versions of the subsymbolic action-planning mechanism that attempt to pursue plans from and towards unexplored regions of the robot's perceptual space. In this way we avoid the representational dead-end problems that (Bugmann, 1997) discusses.

In order to search for plans, we used the concept of reaction-diffusion dynamics, introduced by (Turing, 1952). The use of reaction-diffusion dynamics to accomplish a search through a *symbolic* search space was proposed by (Steels, 1988), and has been used frequently in Artificial Intelligence for searching in artificial worlds. Applications such as the Tower-of-Hanoi problem or abstract path planning are presented, for instance, in (Fleuret and Brunet, 2000) and (Fomin et al., 1996). In contrast to this work, we use reaction-diffusion dynamics for searching through a *subsymbolic* search space.

1.2 Motivation

The subsymbolic action-planning mechanism that we studied in (Pisokas and Nehmzow, 2003) has been proved capable of pursuing plans without external specification of world models and of associations among preconditions, motor actions and postconditions. However, this mechanism was unable to pursue plans from and towards unexplored regions of the perceptual space of the robot.

The hypothesis we formed and which motivated the research presented on this paper can be phrased as: "Plans can be approximate; formed as an assembly of known actions and a few educated guesses". In order to investigate this hypothesis we developed two different subsymbolic planners and then compared them with the original one which we have presented in (Pisokas and Nehmzow, 2003).

The first planner exploits the topology preserving nature of self-organising feature maps while the second planner incorporates radial basis function neural networks for predicting the sensory perception of the robot after performing a motor action. In both approaches the planning mechanism tries to predict what will happen if the robot attempts to move through an unexplored area of its perceptual space.

2. The Subsymbolic Action-Planning Mechanism

2.1 Architecture

The architecture of the subsymbolic action-planning mechanism is depicted in figure 1.

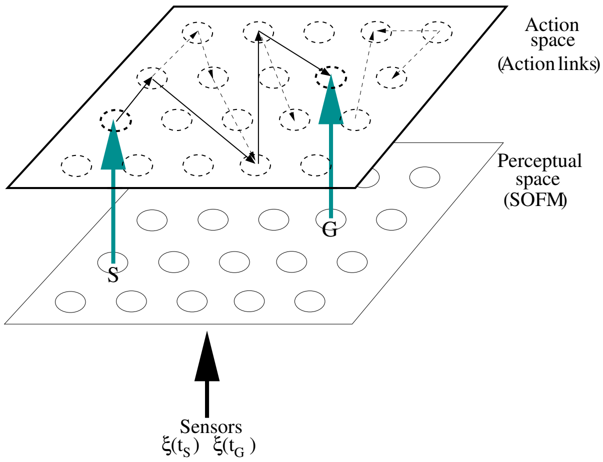

Figure 1: The architecture of the subsymbolic action-planning mechanism. A self-organising feature map clusters sensory perception (perceptual space layer). Action-links, between pairs of perceptual clusters, are stored in a second layer (action space layer). The 'goal' and 'start' sensory inputs are denoted $\xi(t_G)$ and $\xi(t_S)$ respectively; while the corresponding perceptual clusters are denoted with the letters 'G' and 'S' respectively. The result of planning is a list of action-links that link the perceptual clusters of start and goal locations.

There are two layers: a perceptual space layer and a action space layer. In the perceptual space layer a self-organising feature map (Kohonen, 1984) is used for clustering sensory perception. In the action space layer the clusters of the first layer are linked by motor actions available to the robot. To determine 'paths' between start and goal perceptions, reaction-diffusion dynamics are used. The details of the mechanism are discussed in (Pisokas and Nehmzow, 2003, Pissokas and Nehmzow, 2002), while an overview is given in section 2.2.

[1] *Tacit knowledge* is the kind of knowledge that the system cannot reason about. A form of tacit knowledge is the procedural code in a program.

2.2 Operation

There are two operation phases: an exploration phase and an application phase. Initially, during the robot's exploration phase, the robot is left to explore its environment executing random motor actions, from its behavioural repertoire. Each action takes the robot from an initial sensory perception to a resulting sensory perception, collected as a perception-action-perception triple. The acquired sensory perceptions are fed into the SOFM and training it leads to more and more clearly defined clusters. When this phase is completed, raw sensory perceptions are 'encoded' in a topological fashion by the SOFM.

After the representation of perceptual space is completed, the robot uses the perception-action-perception triples, acquired while exploring its environment, to train the action space layer. Motor actions are therefore linked to perceptions encoded on the SOFM, and are consequently entered into the mechanism as directional action-links between initial and final perceptual clusters labelled with the corresponding action (see figure 1). In this manner, the robot develops a representation of perception-action-perception triples throughout the exploratory phase. Clearly, which triples are entered into this representation is dependent on the actual exploratory trajectory taken and the duration of exploration.

Once the world representation is established, it is relatively straightforward to search for complete paths between the current (initial) sensory perception and an externally specified goal perception. The search mechanism we use is one of reaction-diffusion dynamics. Placing an imaginary 'scent' at the goal location and spreading it along all existing links pointing towards the goal location, the 'scent' will eventually reach the start location in perceptual space, indicating a complete path from the start to the goal perceptual cluster, if any exists.

Once the robot has executed the first motor action of the plan, the planning algorithm, described above, is invoked again to make a new plan from the current sensory perception to the goal.

2.3 The Problem

A problem arises when either the current physical location or the goal location (or both) have never have been visited before, meaning that no entry for them is found in the mechanism. In such a case the original planner is not able to pursue a plan, and terminates.

In practice this occurs frequency because it is time consuming for robots to explore their environment exhaustively. Therefore, it is essential to have a planning mechanism capable of pursuing plans from and towards unexplored areas of the environment.

3. Extending the Original Planner

In order to enable the subsymbolic planner to pursue plans through unexplored areas of the environment we experimented with two approaches. The first approach exploits the topology-preserving nature of self-organising feature maps and the second approach incorporates radial basis function neural networks.

3.1 Exploiting the Topology-Preserving Nature of SOFM

We initially tried to exploit the topology-preserving nature of the SOFM assuming that similar perceptions correspond to neighbouring areas of the environment. In this approach, if there is no path from the start perceptual cluster to the goal perceptual cluster then an attempt is made for finding a path that links any two clusters from the neighbourhood of the start cluster and of the goal cluster. Consecutive trials are made each with increasing neighbourhood size until a path is found or the maximum neighbourhood size is reached. The first two steps of the searching procedure, in the case that no complete path from the start to the goal exists, are graphically depicted in figure 2.

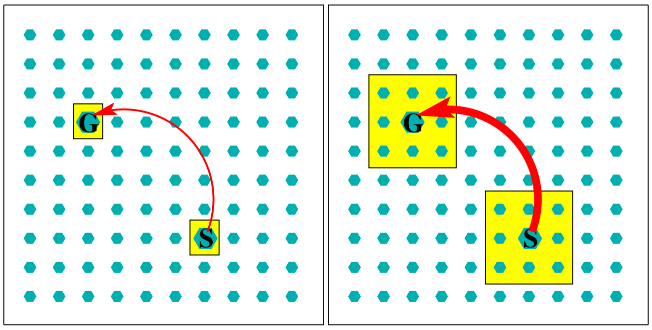

Figure 2: Graphical representation of the neighbourhood size (grey areas) increase during subsequent trials to find a path from the start cluster (denoted as 'S') to the goal cluster (denoted as 'G'). On the left a path only from the start to goal is attempted while on the right both the start and goal have neighbourhoods of radius equal to 1 and a path between the two neighbourhoods is attempted.

3.2 Incorporating RBF Neural Networks

The subsymbolic action-planning mechanism discussed in section 2.1 is only able to use information about perception-action-perception triples that have explicitly been encountered before, it is unable to generalise over the robot's experience.

However, the changes in perception as the robot performs a specific action will commonly be the same, whether a *specific* stimulus has been perceived before or not. For instance, when approaching a specific wall, perceptions will change in some specific way (features will increase in size, and move away from the centre). This would be true, however, whether the robot approached a known wall or an unknown one — our aim was therefore to use learning predictors that would be able to make predictions about perceptual change as the robot performs specific motor actions. The learning predictors were implemented using neural networks.

Due to the limited amount of data that we could obtain in reasonable amount of time[2] we decided to use radial basis function (RBF) neural networks (Lowe and Tipping, 1996, Bishop, 1995). We use four RBF neural networks one for each motor action.

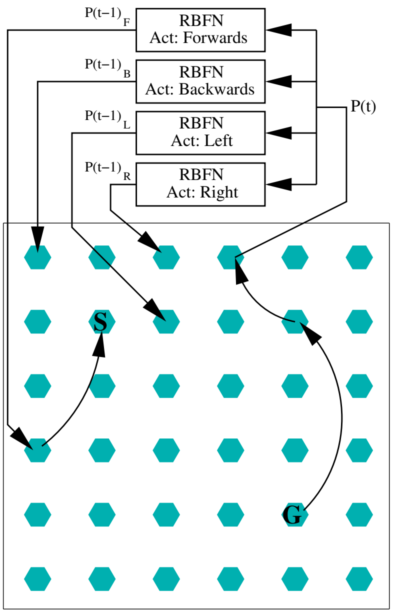

Figure 3: The architecture of the subsymbolic action-planning mechanism with RBF networks employed as perception predictors. A radial basis function network (one network for each of the four actions 'forward', 'backward', 'left' and 'right') associates the 16 sonar sensor signals at time t with the 16 sonar signals perceived at the previous time-step $t-1$. When there is no known path from start to goal RBF networks are employed to complete missing links. The arrows on the figure represent the direction of search (their direction is opposite of the effect and order that the motor actions will have, because the 'scent' propagates from the goal to the start, see section 2.2).

During searching for a path whenever the 'scent' reaches a perceptual cluster, of the SOFM, which has no links going towards it, the four RBF networks are employed to give four links to other perceptual clusters. After that the search continues as usual.

[2]Such problems do not exist with simulated robots where large amounts of data can be collected quickly, for example see (Baldassarre, 2001b).

4. Experiments

4.1 Experimental Procedure

Experimental Setup

For conducting the experiments we used a Magellan Pro mobile robot (figure 4) developed by iRobot Corporation. As sensor inputs to the subsymbolic planning

Figure 4: The Magellan Pro mobile robot, it has 51 sensors: 16 ultrasonic sonars, 16 infrared distance sensors, 16 bumpers, 2 wheel encoders and a colour camera.

mechanism we used the 16 ultrasonic sonars which are evenly placed around the robot's body. The behavioural repertoire of the robot was composed of four primitive motor actions (i.e. 'move forwards 50 cm', 'move backwards 50 cm', 'turn left 40°', and 'turn right 40°'). If the robot encounters an obstacle at a range of less than 20cm it stops and moves in the opposite direction for 10 cm or 10° if it was a translational or rotational action respectively.

The experimental arena used for the experiments is depicted in figure 5. It is surrounded by walls (melamine faced chipboard panels) and there are some objects placed on the sides.

Experimental Method

Initially the robot was left to explore its environment and then the three planning mechanisms were trained using the acquired data (see section 4.2). Next we performed the comparison experiments (see section 4.3). For these experiments the robot's task was to move from a starting position and orientation to another specified goal position and orientation. For this purpose the robot was initially placed at a 'goal' location (to obtain the sensory perception of the goal), then lifted to an arbitrary 'start' location and left to determine a sequence of motor actions that would take it to the goal. We selected five such tasks randomly, with different locations and

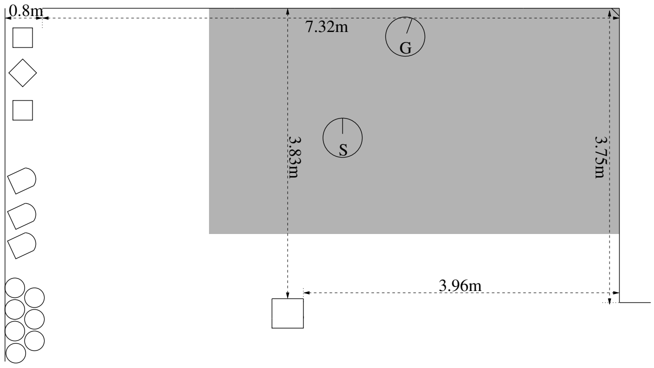

Figure 5: The arena used during the experiments with the Magellan Pro robot. Lines represent walls (made of melamine faced chipboard panels) and the various shapes on the left represent other objects placed on the environment. The grey area designates the area where the start and goal positions of every task were located. An example of start and goal locations is depicted.

orientations. As an example, the start and goal locations of the robot for the first task are depicted in figure 5. For all tasks both the start and goal positions of the robot were lying within the grey area of figure 5. We selected this area because its simple structure allows us to easily specify the position of the robot with distances and angles in respect to the walls. As a result, areas of the environment that give the same sensory perception (aliasing) can be easily identified by the experimenter. This allows us to concentrate on the study and comparison of the three subsymbolic planers.

For comparing the performance of the three subsymbolic planners (the original-planner, the neighbourhood-planner that exploits the topology-preserving nature of SOFM, and the RBF-planner that uses RBF neural networks to predict the robot's perception) we performed 22 runs with each planner for each of the 5 tasks.

In order to avoid bias within the experimental results arising from factors, such as, different charge levels of the battery and environmental changes, the experiments were conducted in sequences consisting of one run of each of the planners.

Performance Criterion

As performance criterion we used the failure rate of the planner. Failure rate was defined as the number of incomplete runs over the number of trials for a certain task. An incomplete run occurs when the robot stops, not being able to plan a path from the current position to the goal or if the time used for accomplishing the task has exceeded 5 minutes.

4.2 Exploration Phase

The exploration phase was performed only once before starting the comparison experiments. The robot explored its environment performing 2823 randomly selected actions out of its behavioural repertoire (approximately 700 repetitions of each of the 4 motor actions available to the robot), and acquired perception-action-perception triples. In cases where the robot was moving far away from the walls towards other areas of the lab, we stopped the robot and moved it to a randomly selected orientation and position between 0.5m–1.5m from the closest wall. This happened 16 times out of the 2823 exploratory actions that the robot performed.

Training the SOFM

The acquired perception-action-perception triples were used to create (i) a subsymbolic representation of perceptual space, using the self-organising feature map (SOFM), and (ii) a representation of perception-action-perception associations. The self-organising feature map that we used was a two dimensional toroid with size 30x30 units and 16 inputs. The sensory perception inputs were presented to the SOFM 50 times during training. The weight vectors were not normalised. The learning rate a starts from value 1 and decreases to 0 in steps of $\frac{1}{Training\ Cycles} = 0.02$ after each training iteration. Therefore, after completing 50 training cycles a becomes 0 and the training stops. The weights update neighbourhood radius N_c starts from $N_c = 15$ and decreases to 1.

The precondition-action-postcondition associations were stored as a list of action-links (links labelled with the action taken) between pairs of perceptual clusters of the SOFM. Action-links occurring more than once were eliminated to only one in order to avoid searching a path more than once.

Training the RBF Neural Networks

As we mentioned previously we used one RBF neural network for each motor action of the behavioural repertoire of the robot. Each neural network had 16 input nodes, 16 output nodes and Gaussian hidden nodes. For training the four RBF networks we separated the dataset (2823 samples) to four subsets each containing only the samples corresponding to one motor action. Each subset (perception-action-perception triples) was separated into a training set (90% of the subset) and a validation set (10% of the subset).

Subsequently the centres of the hidden (RBF) units of the network were initialised by randomly drawing vectors from the corresponding training data subset. After initialisation, the centre vectors were updated for 50 iterations through a competitive learning process, using equation 1.

$$\mathbf{c_w}(t+1) = \mathbf{c_w}(t) + \eta_c(\mathbf{i} - \mathbf{c_w}(t)), \tag{1}$$

with $\mathbf{c_w}$ being the centre vector with the smallest Euclidean distance to the input vector, and $\mathbf{i}$ the input vector. η_c is the learning rate (initialised as $\eta_c = 0.05$, then reduced by 0.001 in each iteration).

Once RBF weights were fully trained, the σ value of each RBF unit was adjusted such that σ became half the Euclidean distance to the nearest RBF unit centre.

The weights of the output units were initialised to random numbers uniformly distributed between -0.0001 and +0.0001. They were then trained using the gradient descent rule given in equation 2.

$$w_{kj}(t+1) = w_{kj}(t) + \eta_o(\tau_j - o_j)\varphi_k, \tag{2}$$

with w_{kj} being the weight from the output of the hidden unit k to the output unit j, $\eta_o = 0.01$ the learning rate for the output layer, τ_j the target of output j, o_j the value of output j, and φ_k the output of the RBF unit k.

In order to select appropriate sizes of the RBF networks we trained several networks of different number of hidden units, and we compared their accuracy of prediction. Figure 6 depicts the prediction RMS error for each of the motor actions for different sizes of RBF neural network.

As depicted in figure 6 the prediction error starts converging for networks with more than 50 hidden units. It is clear that even the minimum values of the validation RMS error are relatively high and this is a consequence of the limited amount of training data. It is not easy, nor necessary, to select an optimal size of network because the training data-set is limited and probably does not cover the robot's perceptual space sufficiently. Fine tuning the size of the network for the given data-set will not be of much use because in a subsequent exploration phase the robot might encounter different areas of the environment resulting in a different data-set which corresponds to a different optimal network size. Therefore, we selected networks that combine converged prediction error with a minimal number of hidden units, in order to keep the computational complexity of the system low.

As a result, we used a neural network with 250 hidden units for each of the motor actions 'move forwards' and 'move backwards' and 200 hidden units for each motor action 'turn left' and 'turn right'. In figure 7 the training and validation error vs. the training iteration is depicted for each one of the RBF neural networks. The stopping criterion was determined to stop training if the validation error increased for 10 consecutive iterations or if there was no improvement for 200 iterations.

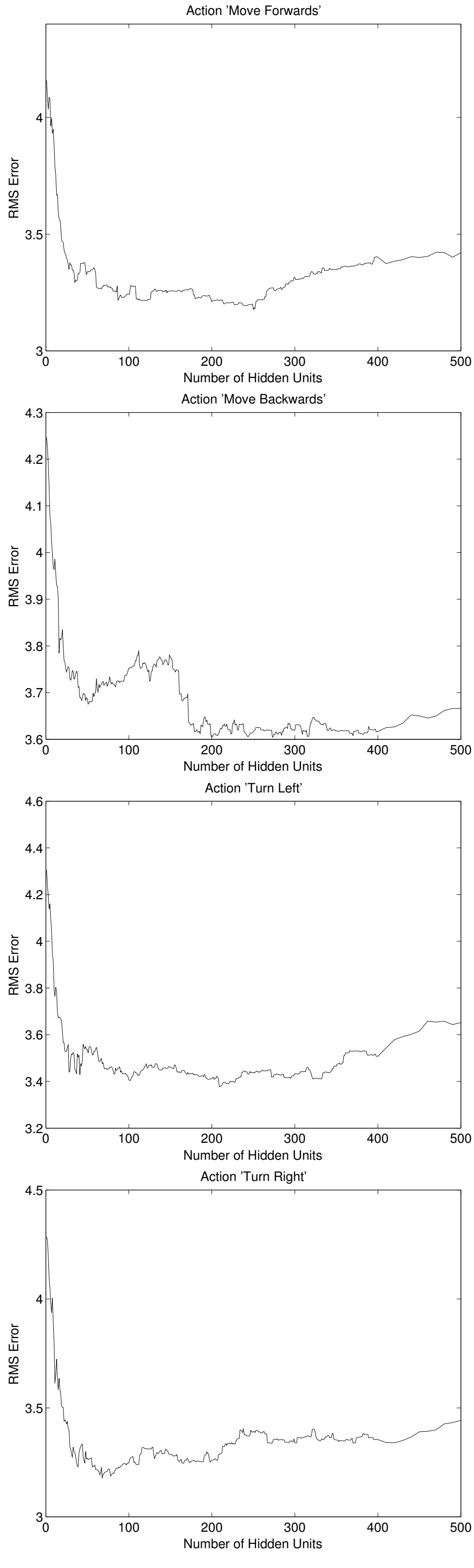

Figure 6: The validation RMS error for different numbers of hidden (RBF) units. One graph per motor action is displayed.

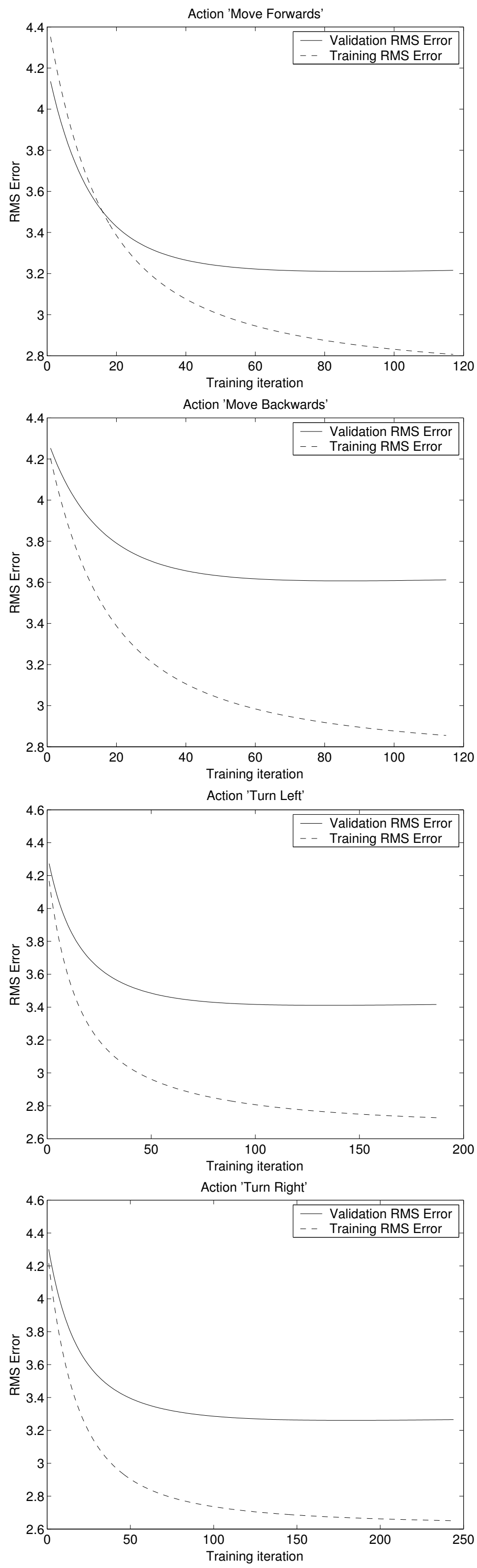

Figure 7: The progression of the RMS error for the training and validation data-sets during training is depicted. One graph for each of the four motor actions is displayed.

4.3 Experimental Results

Failure Rate

In order to study the performance of each planner we measured, and depict in table 1, the failure rate for each of the three planners. A graphical representation of the same data is displayed in figure 8.

Task	Failure Rate		
	Original (%)	Neighbourhood (%)	RBF (%)
1	36	9	14
2	27	18	18
3	59	18	27
4	91	32	55
5	14	0	14

Table 1: Failure rate measured as percentage of incomplete runs per 22 experimental runs. Results for the 3 versions of the planning mechanism are presented.

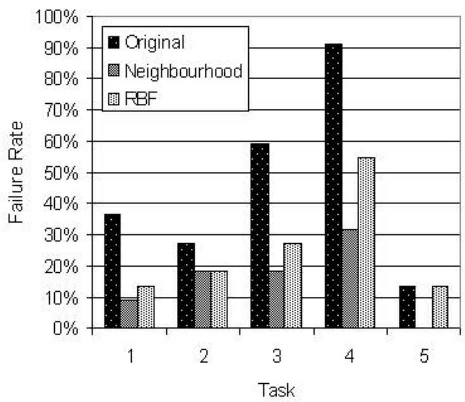

Figure 8: Graphical representation of failure rate measured as percentage of incomplete runs per 22 experimental runs. Results for the 3 versions of the planning mechanism are presented.

As can be seen, the original planner in all cases results in the maximum failure rate because, as soon as the robot considers a sensory perception that has no known path to the goal, it stops.

Comparing these results with those of the neighbourhood-planner, we realise that the neighbourhood-planner generates far less failures (incomplete runs), and those are due to exceeding the available time for the robot to reach the goal (5 minutes).

Finally, the RBF-planner in most cases gives smaller failure rate in respect to the original-planner because it tries to guess actions when there is no known path to the

Task	Number of Motor Actions			
	Statistic	Original Search	Neighbourhood	RBF
1	Mean	10.1	14.3	10.4
	SD	5.3	11.6	5.8
2	Mean	10.6	15.2	10.7
	SD	6.4	13.7	10.1
3	Mean	9.7	19.1	9.7
	SD	8	17.1	8.1
4	Mean	19.5	34.4	31.7
	SD	4.9	18.3	16.9
5	Mean	6.4	10.2	8.5
	SD	3.7	7.2	7.1

Table 2: Mean number of motor actions per experimental run for accomplishing each task (move from start to goal) and corresponding standard deviation. The mean and standard deviation have been calculated only for the complete runs of each task.

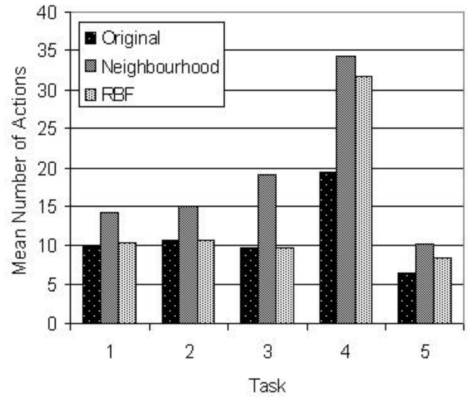

Figure 9: Graphical representation of mean number of motor actions per experimental run for accomplishing each task (move from start to goal).

goal and thus does not stop at the first difficulty. However, the failure rate for the RBF-planner tends to be higher than that of the neighbourhood-planner. This is an expected result because the neighbourhood-planner uses an increasing neighbourhood size until a path is found or no larger neighbourhood is possible. This practice increases the chances of reaching the goal by trying for the entire available time (5 minutes).

Thus far, it is clear that both the neighbourhood and the RBF planners result in a smaller failure rate than that of the original planner.

Mean Number of Actions

As a next step, we studied the number of actions required to accomplish the task for each of the planners, in order to determine whether there is a pattern. In table 2 the mean and the standard deviation of the number of actions are depicted for each experimental task. A graphical representation of the same data is displayed in figure 9.

These results illustrate that the original-planner in all cases results in a minimum number of actions in respect to the other two planners, because it never detours from what it knows searching for a path. The relatively small standard deviation supports this claim. This is an "optimal or nothing planning" approach as done by classical planning systems like STRIPS.

On the other hand, the neighbourhood-planner in all cases results in a maximum number of actions because often the robot, not having a known path to the goal, detours from the explored areas of its perceptual space trying to reach the goal. However, sometimes the neighbourhood-planner results in random search, because the perceptual neighbourhood does not always correspond to topological neighbouring areas in the environment. The occurrence of random search is suggested by the large standard deviation.

Finally, the RBF-planner produces a number of actions which lie between those of the other two planners. This is reasonable because in some cases the robot detours from the explored areas of its perceptual space to search for the goal. The difference to the neighbourhood-planner is that the RBF-planner attempts only educated guesses, exploiting the generalisation capabilities of artificial neural networks. As a result, the number of actions is much smaller than that of the neighbourhood-planner, indicating that the RBF-planner produces more successful guesses. Additionally, the standard deviation corresponding to the RBF-planner is much smaller suggesting that it produces educated guesses rather than random search.

We should mention that there is no statistically significant difference among the results obtained with the three planners (t-test, p=0.05). This is because using any of the three planners the robot has the chance to reach the goal without any guesses resulting to a minimum number of actions, thus the distributions of the number of actions corresponding to the three different planners overlap significantly.

However, there is a clear tendency for the RBF-planner to produce a number of actions smaller than that of the neighbourhood-planner and, in most cases, similar to that of the original-planner.

In conclusion, the experimental results indicate that the RBF-planner provides a good compromise between low failure rate and small mean number of actions, when compared to the other two planners.

5. Discussion

This paper addresses the question of how a mobile robot can autonomously determine task-achieving sequences of actions without using pre-supplied symbolic knowledge, but generating novel plans using previously *acquired* subsymbolic representations.

In particular, we presented two extensions of the subsymbolic action-planning mechanism studied in (Pisokas and Nehmzow, 2003), which was not capable of pursuing plans from and towards unexplored regions of the perceptual space of the robot.

In a first attempt to overcome this problem we tried to exploit the topology-preserving nature of the SOFM. In this way, if there is no known path from the start cluster to the goal cluster, a path between the neighbourhoods of those clusters is attempted. However, this method proved not to be reliable because a small change in perceptual space does not necessarily imply a small distance in topological space. Motivated by this problem we developed another version of the subsymbolic planning mechanism which employs RBF neural networks. This planner is capable of generalising over the perceptual space of the robot and of pursuing plans through unexplored areas of the perceptual space of the robot.

The experimental results illustrate that both the neighbourhood-planner and the RBF-planner reduced the failure rate with respect to the original-planner. Therefore, allowing for approximate plans increases the probability of reaching the goal. Of course the predictions are not always correct and, thus, for both the neighbourhood and the RBF planners an increased number of motor actions are required to accomplish a task.

Having compared the three versions of the subsymbolic planner we draw a new hypothesis that will lead our future research. Our hypothesis is that the RBF-planner will result in less failures and much shorter paths compared to the neighbourhood-planner when operating in environments where small changes in the robot's physical location often cause large changes on its sensory perception and vise versa. This kind of environment is common in the real world. In order to explore our hypothesis we have developed an environment richer in perceptual clues than those we have used until now. Additionally, we have increased the sensor apparatus of the robot by adding inputs from infrared range detectors and a colour camera.

Acknowledgements

John Pisokas is sponsored by IKY (State Scholarship Foundation of Greece), whose support is gratefully acknowledged. We would also like to thank Gary Ireland and Hugo Vieira Neto for their useful commends.

References

Arkin, R. (1998). *Behavior-Based Robotics*. MIT Press.

Baldassarre, G. (2001a). A planning modular neural-network robot for asynchronous multi-goal navigation tasks. In *Proceedings of the 2001 Fourth European Workshop on Advanced Mobile Robots - EUROBOT-2001*, pages 223–230.

Baldassarre, G. (2001b). *Planning with neural networks and reinforcement learning*. PhD thesis, University of Essex.

Bishop, C. (1995). *Neural Networks for Pattern Recognition*. Oxford: Clarendon Press.

Bugmann, G. (1997). A connectionist approach to spatial memory and planning. In Landau, L. J. and Taylor, J. G., (Eds.), *Basic Concepts in Neural Networks: A survey*, In the Series: Perspectives in Neural Networks, chapter 5, pages 109–146. Springer, London.

Fikes, R. and Nilsson, N. (1971). STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2:189–208.

Fleuret, F. and Brunet, E. (2000). DEA: An architecture for goal planning and classification. *Neural Computation*, 12(9):1987–2008.

Fomin, T., Rozgonyi, T., Szepesvári, C., and Lörincz, A. (1996). Self-organizing multi-resolution grid for motion planning and control. *International Journal of Neural Sciences*, 7:757–776.

Gat, E. (1992). Integrating planning and reacting in a heterogeneous asynchronous architecture for controlling real-world mobile robots. In *Proceedings AAAI-92*, volume 4(4), pages 809–815.

Harnad, S. (1990). The symbol grounding problem. *Physica D*, 42:335–346.

Kohonen, T. (1984). *Self-Organization and Associative Memory*. Springer-Verlag.

Lowe, D. and Tipping, M. E. (1996). Feed-forward neural networks and topographic mappings for exploratory data analysis. *Neural Computing and Applications*, 4:83–95.

Malcolm, C. and Smithers, T. (1990). Symbol grounding via a hybrid architecture in an autonomous assembly system. *Robotics and Autonomous Systems*, 6(1&2) (Special Issue – Designing Autonomous Agents).

Nehmzow, U., Smithers, T., and Hallam, J. (1991). Location recognition in a mobile robot using self-organising feature maps. In Schmidt, G., (Ed.), *Information Processing in Autonomous Mobile Robots*. Springer Verlag.

Nilsson, N. J. (1984). Shakey the robot. Technical Report *323*, SRI International.

Pisokas, J. and Nehmzow, U. (2003). Experiments in subsymbolic action planning with mobile robots. In *Proceedings of AISB-03 Convention: Cognition in Machines and Animals, Adaptive Agents and Multi-Agent Systems Symbosium*, pages 80–87.

Pissokas, J. and Nehmzow, U. (2002). Robot action selection through subsymbolic planning. Technical Report *CSM-374*, University of Essex.

Steels, L. (1988). Steps towards common sense. In *Proceedings ECAI 1988*, pages 49–54.

Turing, A. (1952). The chemical basis for morphogenesis. *Phil. Trans. Roy. Soc.*, 37:129–152.

Zeller, M., Sharma, R., and Schulten, K. (1996). Topology representing network for sensor-based robot motion planning. In *Proceedings of the 1996 World Congress on Neural Networks*, pages 100–103. INNS Press.