Design of a biologically inspired navigation system for the Psikharpax rodent robot

K Caluwaerts, C Grand, S N'Guyen, L Dollé, A Guillot, M Khamassi
Institut des Systèmes Intelligents et de Robotique, ISIR CNRS UMR7222

Abstract—**This work presents the development and implementation of a biologically inspired navigation system on the autonomous Psikharpax rodent robot. Our system comprises two independent navigation strategies: a taxon expert and a planning expert. The presented navigation system allows the robot to learn the optimal strategy in each situation, by relying upon a strategy selection mechanism.**

Keywords— **Robotics, place cells, biologically inspired navigation, Q-learning, strategy selection**

I. Introduction

A BIOLOGICALLY inspired navigation system for the mobile rat-like robot nicknamed Psikharpax is presented, allowing for self-localization and autonomous navigation in an initially unknown environment. Parts of the model (e.g. the strategy selection mechanism) have been validated before in simulation, but have now been adapted to a real robot platform.

This article presents our work on the implementation of two independent navigation strategies and a strategy selection mechanism. We show how our robot can learn to choose the optimal strategy in a given situation using a Q-learning algorithm.

II. Background

A. The Psikharpax robot

The Psikharpax robot[1] is a so-called Animat[2], relying upon a bottom-up approach, taking all real-life properties and limitations into account. A rich set of sensory equipment is available to the robot.However, in this work we mainly rely upon visual input, but extensions have been anticipated.

This article is the first to report on the new version of Psikharpax (v2). The robot is now slightly bigger (about 50cm in length), has been equipped with a paw and its head can now be lifted.

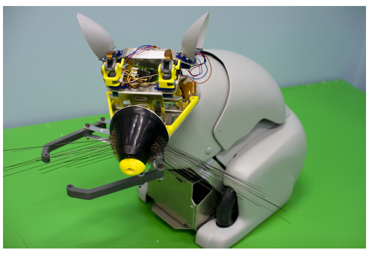

Fig. 1. The Psikharpax robot

B. Overview

The theoretical foundation of our work comes from [3]. A premise of this model is that animals possess multiple independent navigation strategies[4], which can *cooperate* or *compete*. It is not yet clear how these strategies are mediated and the model we implemented provides a simple mechanism that allows the robot to learn the optimal strategy in each state, based on Q-learning in continuous space (state and action).

The first part of this paper gives a brief technical overview of the platform. The theoretical foundation of our work was verified by Dollé et al. [3] in simulation, based on almost perfect sensory input and simulated grid cells[5]. Therefore, the second part of this paper presents the equivalent navigation strategies for the real robot. The last part of this paper covers the strategy selection mechanism.

Our system also has a means of estimating its allocentric bearing, similar to [6].

III. Strategies

A. Taxon strategy

Our system comprises two independent strategies. The first strategy, the taxon expert, learns to associate visual cues with actions (model-free) and is based on the equations found in [3]. The taxon strategy in this work is used to provide a guiding expert by using the relative position of the goal as input (given by a ceiling camera), but distorted by a Gaussian noise increasing quadratically with the distance to the goal. This provides the robot with useful guiding only when near the goal.

B. Planning expert

The second strategy is a planning expert, similar to the one found in [7]. The planning expert can take any type of orientation-independent information as input and associates inputs by recruiting neurons in an incremental way. Connections between places are learned with Hebbian-like rules.

To plan a path, the most active node in the map is used as the current location (winner-take-all). Through a diffusion mechanism, the goal value is propagated from the goal(s) to each node. This gives the robot a means of computing the shortest path to the closest goal in terms of the number of intermediate nodes. When the path nodes are known, a path is planned in the odometric reference frame, allowing the robot to navigate for a while without reliable input.

In our current model, the input of the planning expert is solely based on visual place cells information. At the lowest level of the visual system, the input is processed by Bio-Inspired Perception System hardware processors [8]. The output from this layer are the salient objects in the environment[1] and their properties. At the

[1]The environment measured $\sim$ 2.5m by 2m and contained only extra-maze cues at distances up to 1m from the border of the envi-

second layer, the detected objects' properties are neurally coded. An additional layer integrates the visual perception to reproduce an approximate 360 degree vision while moving, as our robot only has a 60 degree field of view.

To estimate the quality of the current visual place cells activation, we keep track of a set of trust neurons.

To create non-directional place cells from the visual integration layer, we sum this information over all horizontal directions, creating a feature map.The Growing Neural Gas[9] algorithm was used to construct place cells.

By fixing the maximal number of neurons, the number of zones can be chosen at wish. In fig. 2, we show a spatial plot of the activation of 4 sample place cells covering the environment. A binary activation (BMU of the GNG) is used to project onto the topological map. A smoother activation function, activating also the neighboring neurons of the best matching unit, is used as input to the gating network.

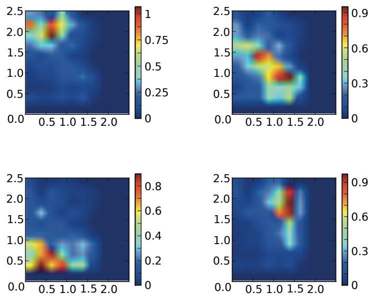

Fig. 2. Heat map showing the activation for 4 place cells as a function of the location of the animat.

IV. Strategy selection

A. *Gating network*

In our system, all experts work and learn in parallel. A Q-learning algorithm based on the simulation model from Dollé et al. [3] is implemented to allow the robot to associate a state with an optimal strategy. Contrary to the simulation model, we only used the place cells as input to the gating network. This allowed for dimensionality reduction and easy evaluation of the results.

The gating network computes the gating values $g^k(t)$, one for each strategy k. The Q-values are stored in a matrix $z_j^k(t)$, associating inputs from the place cells with gating values:

$$g^k(t) = \sum_j^{N_{PC}} z_j^k(t) n_j^{PC}(t) \tag{1}$$

We do not adopt the winner-take-all[2] policy from the simulation model to select the winning strategy for the next action. Instead the selection probability of an expert increases with its gating value:

$$\mathrm{P}(\Phi^*(t) = \Phi^k(t)) = \frac{g^k(t)^\zeta}{\sum_i g^i(t)^\zeta} \tag{2}$$

Here $\Phi^k(t)$ is the action proposed by expert k at time t. $\Phi^*(t)$ is the final action proposed by the gating network. Note that this action is not always the executed action, as higher priority strategies (reflexes) can override the gating network. ζ is a parameter increasing with time. For $\zeta = \infty$ our action selection mechanism is equivalent to the one from [3]. For $\zeta = 1$, one obtains the action selection mechanism from [10].

The advantage of introducing some randomness in the action selection is that slower learning strategies can catch up with fast learning strategies when they start to perform better only after a long time. With a winner-take-all strategy, one might have to wait for convergence before a slower learning, but optimal strategy can increase its weights beyond these of a faster learning, but suboptimal strategy.

Learning is sped up by using action generalization and eligibility traces. The equations for these techniques were taken from [3] and we do not repeat them here. A substantial difference lies in the equation to update the eligibility traces. Whereas sensory input is always reliable in the simulation model, it is not in general on the real robot. To incorporate this fact in our system, the eligibility traces are modulated by the trust neurons introduced in III-B:

$$e_j^k(t+1) = n_{conf}^{PC}(t)\Psi(\Phi^*(t) - \Phi^k(t)) r_j^{PC}(t) + \lambda e_j^k(t) \tag{3}$$

B. *Results*

B.1 Planning expert vs. taxon - fixed goal

For this experiment, we connected the planning expert and a pre-trained taxon (to speed up learning) to our system. The goal was to verify if the robot could learn to pick the taxon strategy when close to the goal (where guiding works well), while preferring the planning strategy at a greater distance from the goal. ζ was fixed at 1.

The result is presented in fig. 3. The size of the spots is proportional to $|z_j^0 - z_j^1|$, i.e. the difference between the weights connecting place cell j to both strategies in the gating network. The result is overlayed on an image of the environment.

Around the goal, the robot clearly prefers the taxon strategy, while farther away the planning strategy is more important. Note however that the difference is less pronounced at a greater distance due to the nature of the Q-learning algorithm.

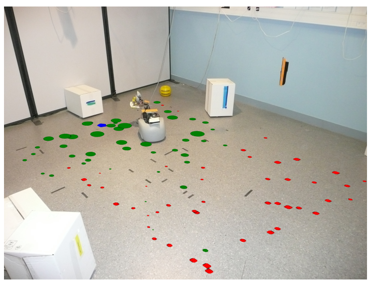

Fig. 3. Planning expert (red squares) vs. pre-trained taxon (green dots). The goal location is shown in blue. The robot has learned to choose the taxon (guiding) expert when close to the goal, while preferring the planning expert when the guiding fails farther away from the goal.

ronment. 13 colorful objects were used as landmarks.

[2] $\Phi^*(t) = \Phi^{\mathrm{argmax}_k(g^k(t))}(t)$.

B.2 Planning expert vs. taxon - two goals

In this experiment, the goal was moved after 1000 steps. Initially, the conditions are the same as in the previous experiment (goal on the left). After 1000 steps the goal is moved to the opposite side of the environment.

The gating network has no way of noticing this, except for the reward signal. As can be seen in fig. 4, the robot learns the new situation but it takes over 8000 steps, while it took less than 1000 steps to learn the initial situation. This is due to the fact that the weights of the taxon around the original goal had already received important reinforcements.

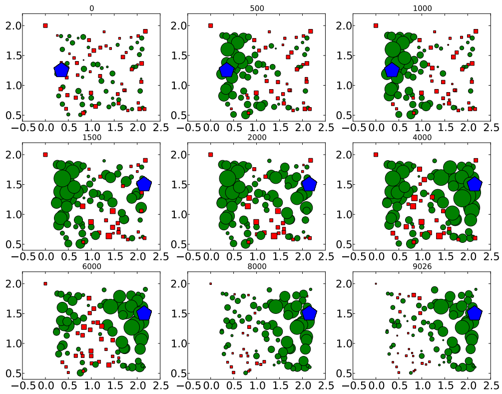

Fig. 4. Planning expert (red squares) vs. taxon (green dots) no context switching. The goal location is shown in blue. The location of the goal is changed after about 1000 steps

To overcome this limitation (inherent to the Q-learning algorithm), we implemented a simple context switching mechanism. Before every step, the gating network decides upon the context it is working in. For this it uses the vector of diffusion values d from the planning strategy. The gating network now stores a set of Q-value matrices $z^i_{i,j}$, and associates a diffusion vector d^i to each matrix. The current context is now chosen as follows (d is the current diffusion vector):

$$v^i = \frac{d \cdot d^i}{\|d\|\|d^i\|} \quad (4)$$

$$z^*_{k,j} = z^{\operatorname{argmax}_i v^i}_{k,j} \quad (5)$$

When $\max_i v^i$ is too small, a new context is recruited. The experiment was repeated with this mechanism (fig. 5). In total 4 contexts were recruited (two transitional). The goal was moved twice, to verify that the robot has learned that the original context has been restored.

Learning times have dropped as the robot does not need to unlearn a previous context, but can recruit (or recall) another one.

V. Conclusions and Future Work

We presented the implementation of a novel strategy selection mechanism allowing an autonomous robot to navigate an initially unknown environment. It was shown that the animat can learn to ignore useless strategies very fast (e.g. an exploration expert after the goal

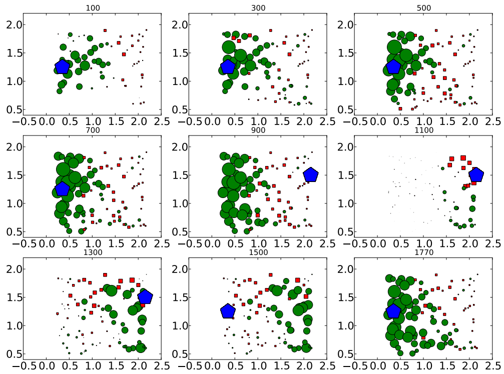

Fig. 5. Planning expert (red squares) vs. taxon (green dots) with the context switching mechanism. The goal location is shown in blue. Learning is now faster as the robot recalls previously learned contexts and new ones. The number of steps is shown at the top.

was found). Furthermore, the robot learned to associate states with optimal strategies even in more complex cases, when for example a local guiding strategy was combined with a global but coarse path planning strategy. By introducing a simple context switching mechanism, the robot can anticipate changes in the environment easily.

We hope to reproduce more complex situations in the future to evaluate the results obtained in simulation and on the real robot.

Acknowledgments

This research was granted by the EC FP6 IST 027819 ICEA Project.

References

[1] J.-A. Meyer, A. Guillot, B. Girard, M. Khamassi, P. Pirim, and A. Berthoz, "The Psikharpax project: towards building an artificial rat," *Robotics and Autonomous Systems*, vol. 50, no. 4, pp. 211–223, Mar. 2005.

[2] S. W. Wilson, "The animat path to AI," in *Proceedings of the first international conference on simulation of adaptive behavior on From animals to animats*, Cambridge, MA, USA, 1990, pp. 15–21, MIT Press.

[3] L. Dollé, D. Sheynikhovich, B. Girard, R. Chavarriaga, and A. Guillot, "Path planning versus cue responding: a bioinspired model of switching between navigation strategies," *Biological Cybernetics*, 2010.

[4] N. Burgess, "Spatial cognition and the brain," *Year In Cognitive Neuroscience 2008*, vol. 1124, pp. 77–97, 2008.

[5] B. Ujfalussy, P. Erős, Z. Somogyvári, and T. Kiss, "Episodes in space: A modeling study of hippocampal place representation," *From Animals to Animats 10*, pp. 123–136, 2008.

[6] Angelo Arleo and Wulfram Gerstner, "Modeling rodent head-direction cells and place cells for spatial learning in bio-mimetic robotics," in *From Animals to Animats*. 2000, pp. 236–245, MIT Press.

[7] N. Cuperlier, M. Quoy, and P. Gaussier, "Neurobiologically inspired mobile robot navigation and planning," *Frontiers in Neurorobotics*, 2007.

[8] S. N'Guyen, *Mise au point du système vibrissal du robot-rat Psikharpax et contribution à la fusion de ses capacités visuelle, auditive et tactile*, Ph.D. thesis, Université Pierre et Marie Curie, 2010.

[9] B. Fritzke, "A growing neural gas network learns topologies," in *Advances in Neural Information Processing Systems 7*. 1995, pp. 625–632, MIT Press.

[10] R. Chavarriaga, T. Strösslin, D. Sheynikhovich, and W. Gerstner, "A computational model of parallel navigation systems in rodents," *Neuroinformatics*, vol. 3, no. 3, pp. 223–241, 2005.