TSRPC '94, Leeuwenhorst, The Netherlands,
June 24-26, 1994

ALICE – Topographic Exploration, Cartography and Adaptive Navigation on a Simple Mobile Robot

Pascal Lefèvre, Andreas Prüß & Uwe R. Zimmer

University of Kaiserslautern - Computer Science Department - Research Group Prof. E. v. Puttkamer
P.O. Box 3049 - 67663 Kaiserslautern - Germany
Phone: ...49 631 205 2624 - Fax: ...49 631 205 2803 - Telex: 4 5627 unikl d
e-mail: uzimmer@informatik.uni-kl.de

A sub-symbolic, adaptive approach to the basic world-modelling, navigation and exploration tasks of a mobile robot is discussed in this paper[1]. One of the main goals is to adapt a couple of internal representations to a moderate structured and dynamic environment. The main internal world model is a qualitative, topologic map, which is continuously adapted to the actual environment. This adaptation is based on passive light and touch sensors as well as on a internal position calculated by dead-reckoning and by correlation to distinct sensor situations. Due to the fact that ALICE is an embedded system with a continuous flow of sensor-samples (i.e. without the possibility to stop this data-flow), realtime aspects have to be handled.
ALICE is implemented as a mobile platform with an on-board computer and as a simulation, where light distributions and position drifts are considered.

1. Introduction

Symbolic- (AI) and functional-decomposition-approaches are the classical techniques to handle mobile robot tasks. Since a couple of years, some basic tasks like collision avoidance, kinematic modelling, reflective navigation etc. are being approximated by much simpler methods like behaviour-based approaches or artificial life resp. genetic/evolutionary algorithms. Some of these works show an encouraging relationship between computational effort and complexity of the handled task (see e.g. [6]).

1. The project ALICE is supported by the EU-project DG XII, F-5 (Teleman)

The still on-going ALICE-project would like to propose some simple concepts for the handling of basic mobile robot tasks, e.g. exploration, world modelling and navigation. The key issue, as understood by the authors so far, is an effective world model, i.e. the quality of adaptation to an environment is measured by the performance of the tasks using this representations and not by the transparency of the produced "map" evaluated by a human operator.

2. Project: ALICE

Based on a continuous flow of passive light and touch sensor-samples from a moderate structured and dynamic environment, ALICE produces a continuously adapted, qualitative, topologic map of sensor situations in order to enable the navigation resp. exploration tasks to perform efficiently.

The ALICE-Project consists of two major parts: The real robot and an equivalent simulation environment. With the help of the operating- and communication-system ALBATROSS [7] we are able to use the same object code on both shells, avoiding cross-compilers and other tools.

The mobile robot ALICE is a round (40 cm in diameter, 20 cm high), fully autonomous platform with an omnidirectional kinematic. 24 whisker-light-sensor pairs are distributed symmetrically at the border of the vehicle and three light-sensors are directed towards the ceiling. Also it is possible to determine a rough internal position by dead-reckoning. The available computer power is limited to one Motorola 68040 processor with 16 MB storage capacity.

The software consists of three major processes:

- *Map-building:*
  Finding an adequate internal representation of the environment as seen by the sensors. Here to-

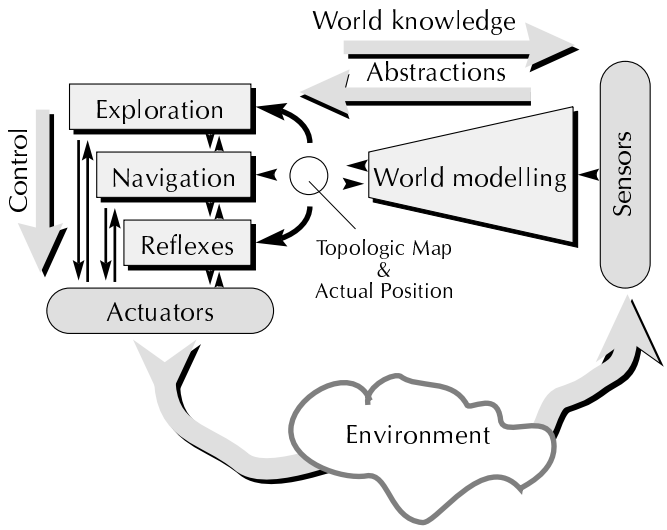

figure 1 : ALICE main-structure

pology-preserving neural network structures are used as the basic concept.

- *Navigation & Pilot:*
  Finding a way (path) to a specified place in the known (mapped) area of the environment and driving along this path, until the robot reaches the target-place or detects a situation which prevents this plan.
  Several reinforcement techniques are used to fulfil this task adaptively (see [9] for details of the navigation technique).
- *Exploration:*
  Finding moves and paths to increase the knowledge, accumulated in the map efficiently.

Although these three components may be seen as different levels of abstraction, they have to be able to operate in parallel. The process of acquiring knowledge by wandering around requires all major components.

3. Qualitative topologic maps

Qualitative topologic maps (QTMs) are an alternative to exact geometric models of the environment (see e.g. [8] as an example of exact geometric mapping based on the data of a laser range finder). Instead of modelling the boundaries of the detected objects, only the sensor-information itself is used to build a "map of sensor-impressions" directly. This concept has already been proposed by Kuipers et al. [1], but the construction process was done by explicit rules, i.e. not by statistical techniques. Our map is constructed by special clustering techniques based on Kohonen's Self-Organizing-Maps (described in the next chapter). QTMs are able to represent significant different sensor-situations and their neighbouring relationships. Therefore similar sensor-situations ("similar" to a representative constellation found in the QTM) can be detected. So the first basic task for each autonomous mobile system can be fulfilled: "Recognize places, where you have been before (without getting exactly the same sensor-measurements)". This task may be summarized by "Qualitative Recognition". On the other hand it is not possible to get positions from this map for an exact recallibration of the internal position. Nevertheless it is possible to correlate the internal position within the boundaries of the granularity of the distinct represented sensor-situations.

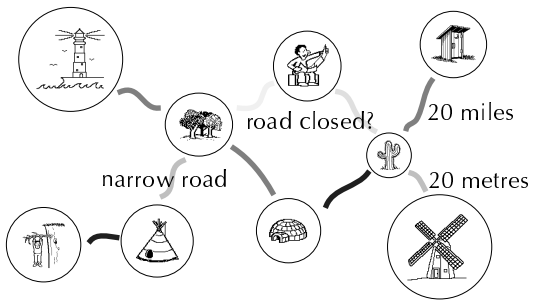

figure 2 : Qualitative topologic map

3-1. Situation vectors

The robot ALICE is wandering around (controlled by an exploration algorithm or a specific navigation task), while producing "situation vectors" in equidistant time intervals. These situation vectors are calculated by concatenating the weighted rough sensor informations, where the weights are approximated by the following assumption. The two situations "In front of an obstacle" and "In touch with an obstacle" should by interpreted as two different situations. Therefore the whiskers should get a relatively high weight. The weights of the light-sensors and position information might be chosen according to the application.

3-2. Clustering method

The situation vectors have to be clustered in order to define areas of similar sensor situations. By adding practicable paths (connections) between nightboured situations, the distinct sensor situations are completed to a topographic map.

The underlying neural network model is basically a Self-Organizing-Map in the variation by Fritzke called Growing-Cell-Structures [2] and with several adaptations and extensions described in [3]. The main idea is to use the neighbourhood connections in a dynamic neural network as a representation of topologic neighbourhood in the environment. The network-processes have to be adapted in a way to avoid not reasonable connections in the topologic interpretation, although these connections would make

sense, if they are only interpreted by the processes for growing and shrinking the network.

3-3. Position correlation

The internal position is corrupted by drift effects and quantization errors. If no compensation of these effects is applied, the robot would (after a certain amount of time) loose contact to the map build up so far, i.e. the robot would not be able to recognize any place resp. sensor-situation. The internal position can only be corrected with the help of the internal world model and at the same time the world model is adapted on the base of the actual sensor situation (i.e. also the internal position). Therefore we can only expect to limit (i.e. not to eliminate) the position error. Furthermore the position correlation can only work, if limitations regarding the length of a path in completely unknown areas are applied.

The position correlation is done by "moving" the internal position slightly towards the position recognized in the internal world model. The "moving factor" has to be adapted according to the actual speed and sampling frequency.

To prevent drifting of the whole model, some well adapted nodes in the network might be chosen as "fixpoints". Therefore nodes which are adapted (confirmed) by many ($n_{fixpoint}$) sensor-samples, are defined as "fixpoints", i.e. their positions will not be updated any longer, whereas the other sensor-information in the world model will be adapted in all cells at any time.

3-4. Optimization criteria

As mentioned above, the main optimization criterion for the internal world model is the performance of the tasks applying the world model. This main criterion might be approximated by several simple measurements:

- *Small number of nodes*: From the set of sufficient world models the most abstract one should be chosen.
- *Short connections and small number of intersections*: The most simple and "flat" structure should be constructed.
- *"Well" adapted representation*: The actual world model should represent the actual environment with a certain accuracy and number of details.
- *Moderate adaptation speed*: The world model should be updated in a continuous manner, in order to be tolerant against statistical errors and noise.

4. Navigation

The navigation module has been split into two layers, which refer to the path-planning task and the problem of translating the planned path into adequate motor actions.

4-1. Path planning

The path planning is being done on the topologic map (graph) by steepest gradient methods and a modified A*-algorithm. Depending on the complexity of the environment both methods produce adequate results. The important criterion is the evaluation speed rather than finding the best way, because small changes in the environment may lead to a failure when trying to drive along a planned path. Therefore the path planning component is employed each time when a plan fails and because we are using inaccurate sensors and dynamic environments this may happen quite often. On the other hand this replanning might be done quickly, because the topologic map is – in the perspective of the path-planner – overspecified and offers a lot of alternative routes.

4-2. Adapting motor actions

In order to navigate a planned path, we have to find a mapping from a pair of nodes in the topologic map (temporary start- and destination-points) to adequate motor commands constrained to a certain accuracy. The kinematics of ALICE are quite simple, because we have an omnidirectional platform and furthermore high accelerations. So we are able to learn this mapping including drift effects, depending on the actual environment as well as on tolerances in the driving/dead-reckoning part of ALICE.

The adaptation is being done (in parallel to the topologic-map-building) with an reinforcement learning scheme (figure 3), where supervised dynamic feature maps (introduced in [2]) are applied in the associative memory module. Later on, the learned transla-

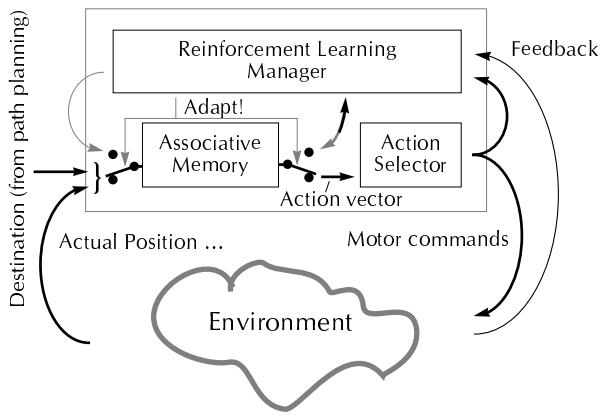

figure 3 : Reinforced learning

tion is applied to areas of the map, which were not available during the learning phase (generalization).

5. Exploration

Three points are important for the construction of an exploration strategy:

- The "amount" of driving in unknown environment-parts must be limited, because otherwise the position correlation will fail.
- Each sensor situation should be "seen" multiple times and from (slightly) different positions, in order to get a statistically stable representation.
- The exploration should be effective, i.e. a certain amount of new information per exploration-time should be gathered.

The reader may notice, that the last point may lead to a contradiction to the first two limitations. All entries in our world model have to be statistically stabled before they are really usable. Therefore we are limited in the exploration-speed, but we get a robust and error tolerant representation.

6. Simulations

The simulations are all done in one environment (figure 4), where three light sources, different kinds of objects, rough and plain surfaces, and a passage (nor symmetrical neither straight) are represented. The light sources are limited by diaphragms like those being used with studio spotlights. Therefore light is

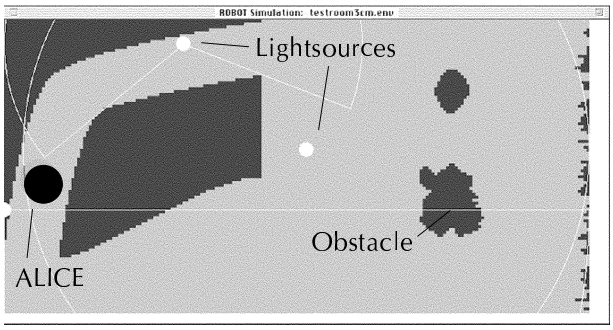

figure 4 : Test-Environment

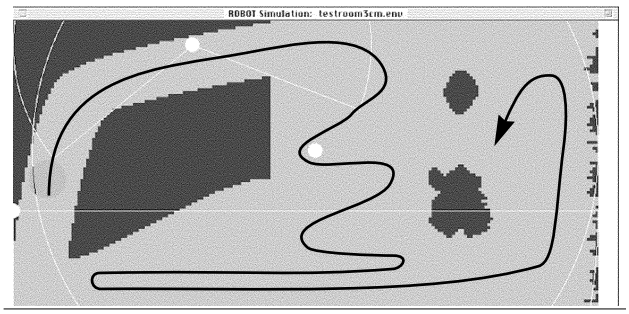

figure 5 : Test-Trajectory

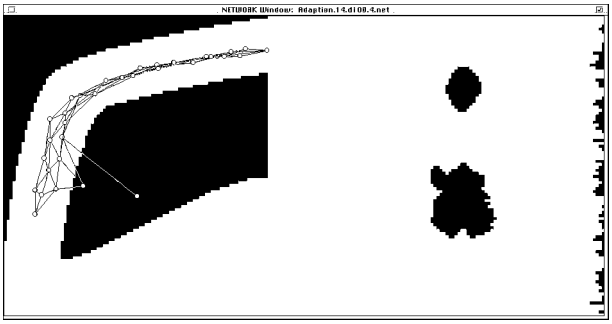

figure 6 : 1/3 of Test-Trajectory passed

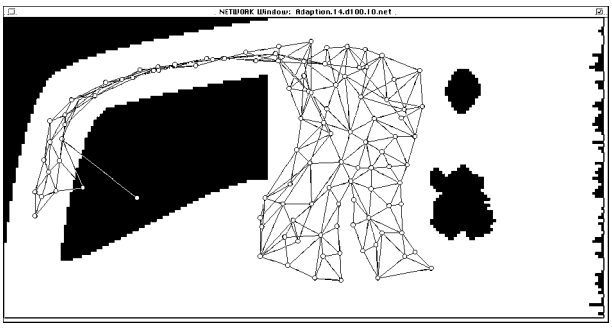

figure 7 : 2/3 of Test-Trajectory passed

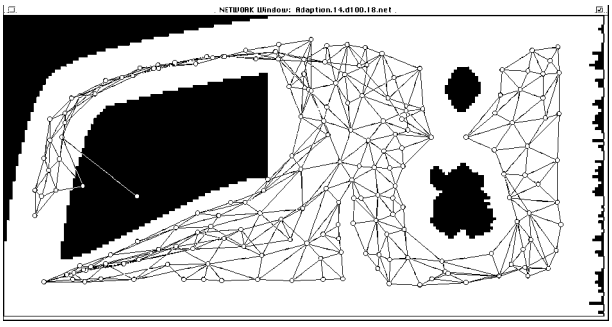

figure 8 : Full Test-Trajectory passed

emitted within a certain angular range only (indicated by the white lines in figure 4).

The described simulation runs are evaluated on the trajectory shown in figure 5. As an exploration strategy, simple random movements are performed in a window wandering along the trajectory.

In figure 6 to 8 the creation of a topologic environment map filling the test environment is shown. Notice that the nodes nearest to the obstacles represent the situations, when ALICE touches the obstacles with its whiskers, whereas nodes in open spaces represents different light situations. The total number of processed input signals was 5254 and the number of nodes in the network was 196. This run can be reproduced on other test-trajectories leading to similar results. Due to the small number of nodes in this examples, we are not forced to limit the total number of nodes. In an open (i.e. not limited) environment, it is necessary to limit the network size according to the available computing power.

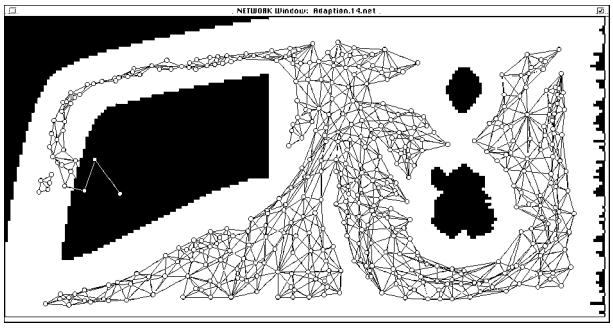

figure 9 : e_{bmu} = 0.01

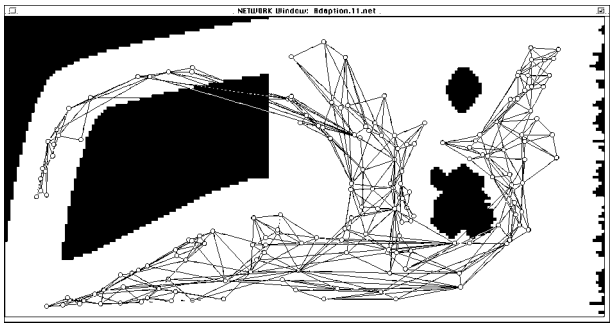

figure 10 : e_{bmu} = 0.1

One important parameter for the adaptation is e_{bmu} (and implicitly $e_{neighbour}$), which determines the adaptation speed as well as the network structure itself. In order to show two typical wrong configured networks, we set e_{bmu} to 0.01, which produces an overfitting of the environment, by inserting too much cells to fulfil the demanded accuracy (figure 9). On the other hand choosing e_{bmu} too high (0.1) results in a fast moving cell structure with chaotic connections and no useful modelling of the environment (figure 10).

The realtime aspects of the simulations might be summarized as follows: The computational effort rises linear with the number of nodes in the network (using parallel techniques it may be reduced to $O(log(n))$). In order to guarantee a certain adaptation speed, the total number of nodes must be limited. Therefore the area which might be explored without segmentation is limited by the "complexity" of the environment (which determines the necessary number of nodes) and of course by the available computing power.

7. Conclusion

The project ALICE is dedicated to a quite new research field, where the combination of basic mobile robot tasks with different neural network learning schemes is proposed. The basic ideas of ALICE (i.e. using dynamic feature maps to implement qualitative topologic maps; adapt the navigator to kinematic and dynamic drifts and restrictions; etc.) have been proven as reasonable in the first tests, where very encouraging simulation-results are produced. The ongoing research is concentrated on tests (and test-strategies) on the mobile platform in different environments as well as finding algorithms to calibrate (adapt) some of the parameters.

References

[1] Benjamin J. Kuipers, Y. T. Byun
A Qualitative Approach to Robot Exploration and Map-Learning
AAAI '87 - Workshop on spatial reasoning and multi-sensor fusion

[2] Bernd Fritzke
Growing Cell Structures - A Self-organizing Network for Unsupervised and Supervised Learning
Technical Report 93-026, International Computer Science Institute, Berkeley, California

[3] Herman Keuchel, Ewald von Puttkamer, Uwe R. Zimmer
SPIN - Learning and Forgetting Surface Classifications with Dynamic Neural Networks
Proceedings of the ICANN '93, Amsterdam, The Netherlands

[4] Teuvo Kohonen
Self-Organization and Associative Memory
Springer-Verlag, Berlin Heidelberg New York Tokyo, ISBN 3-540-12165-X

[5] Teuvo Kohonen
Statistical Pattern Recognition Revisited
Advanced Neural Computers / R. Eckmiller (Editor), Elsevier Science Publishers B.V. (North-Holland), 1990

[6] Ulrich Nehmzow
Experiments in Competence Acquisition for Autonomous Mobile Robots
Ph.D. Thesis - University of Edinburgh 1992

[7] Ewald von Puttkamer, Christopher Wetzler, Uwe R. Zimmer
ALBATROSS - The Communication Scheme as a Key to FulFil Hard Real-Time Constraints
Proceedings of the Euromicro-Workshop on Real-time Systems '92, Athens, Greece

[8] Uwe R. Zimmer
Connectionist Decision Systems for a Visual Search Problem
Proc. of the IIZUKA ´94, Fukuoka, Japan, August 1-7, 1994, Invited paper

[9] Uwe R. Zimmer, Cornelia Fischer, Ewald von Puttkamer
Navigation on Topologic Feature-Maps
submitted to IIZUKA ´94, Fukuoka, Japan, August 1-7, 1994