Neurocomputing, Special Issue
1996

Robust World-Modelling and Navigation in a Real World

Uwe R. Zimmer

University of Kaiserslautern - Computer Science Department
67653 Kaiserslautern - Germany
e-mail: uzimmer@informatik.uni-kl.de
http://ag-vp-www.informatik.uni-kl.de/Leute/Uwe/

This article will discuss a qualitative, topological and robust world-modelling technique with special regard to navigation-tasks for mobile robots operating in unknown environments. As a central aspect, the reliability regarding error-tolerance and stability will be emphasized. Benefits and problems involved in exploration, as well as in navigation tasks, are discussed. The proposed method demands very low constraints for the kind and quality of the employed sensors as well as for the kinematic precision of the utilized mobile platform. Hard real-time constraints can be handled due to the low computational complexity.

The principal discussions are supported by real-world experiments with the mobile robot "ALICE"[1].

Keywords: **artificial neural networks, mobile robots, self-organization, world-modelling, navigation**

1. Introduction & Motivation

Based on the idea that an autonomous robot should have the ability to create and update "relevant" representations or models of its current environment under hard real-time constraints in order to be really useful, a couple of "world-modelling" strategies have been introduced. The attribute "relevant" in this context means: "useful for a certain class of tasks". A wide range of symbolic (AI) and functional decomposition approaches have been already applied for this class of mobile robot tasks (see e.g.[1], [3], [6], [10] or [14]), but the success in real environments depends widely on the quality of the employed sensors. Handling high-speed, high-precision sensors means processing high bandwidth information-streams and results in complex multiprocessor systems and especially in realtime-problems. Furthermore complex computer-systems limit the lower size and weight for the autonomous mobile platform and therefore the range of applications.

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

During the last few years some basic tasks like collision avoidance, kinematic modelling and reflective navigation have been approximated by much simpler methods such as behaviour-based approaches, reinforcement learning methods and others. The idea common to all these techniques is qualitative modelling as opposed to high-precision control. Works utilizing qualitative techniques for navigator-world-modelling include the basic article from Kuipers introducing the term "qualitative map" '88 in [7], the ultrasonic-clustering techniques published by Kurz in multiple articles (e.g. [8]), the expansion of the behaviour-based approach from Mataric published '92 in [11], and the recently published work of Tani based on sensor-sequences rather than on explicit topology [12]. One project handling multiple representation simultaneously, in this case, topology superimposed on the geometrical model, is the HILARE-project published in multiple papers (see e.g. [9]).

Nevertheless, explicit metric world-modelling for navigation tasks is still being employed in most projects. Overcoming the well-known problems of metric mapping, this article introduces a new method of robust, qualitative and topological world-modelling (a term to be defined in the next section) usable for navigation-tasks under real-world constraints.

In order to test the real-environment and real-time assumption, a mobile robot "*ALICE*" was built. The performance of the sensor equipment and mechanics, as well as the available computer power, is limited intentionally to a very low level (i.e. 24 binary whiskers, 24 passive light sensors, 1 Motorola 680x0 CPU, an omnidirectional platform of 40 cm diameter and odometry with up to 25% drift).

2. Topological World-Modelling

The central motivation of **q**ualitative **t**opological world models (**QT**-Models) is the basic mobile robot task: "Recognize places you have seen before!". In this article this task will be approximated by extracting "situations" (i.e. recognized places) together

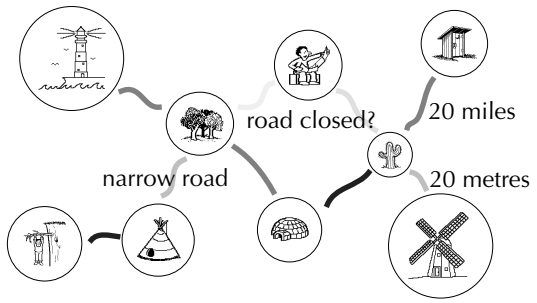

figure 1 : Qualitative topological map

with their topological neighborhood from the current sequence of sensor-samples, rather than modelling the boundaries of the detected obstacles and objects in a metric manner. Assuming a stable situation-recognition-process and a technique for moving between distinct situations, the concept of a qualitative, topological world model (outlined in figure 1) suggests a human-motivated basis for a navigation. The main concept has already been proposed by Kuipers et al. [7], but here the construction process was carried out using explicit rules, not statistical techniques. Therefore the real-world abilities of the Kuipers approach are, in the opinion of the author, limited.

The world model proposed in this article is based on clustering techniques introduced by Kohonen ("self-organizing-maps", [5]) and Fritzke ("growing cell structures", [2]) together with some previously proposed extensions by this research group [4]. Due to a couple of specific autonomous robots-constraints, these structures are modified to cope with realtime-aspects, lifelong learning, "local forgetting", and so on (see section 3 for details).

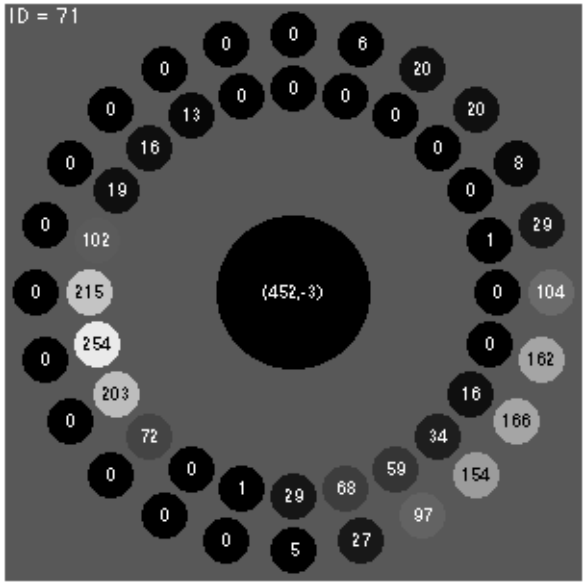

figure 2 : A "situation"

To make the term "situation" in this context more precise, figure 2 shows a typical situation generated in *ALICE*'s test-runs. The inner circle describes the distribution of light-impressions received from particular directions, the outer circle shows the smoothed touch-values originate from contacts with obstacles in various directions. The two-number pair in the centre gives a rough approximation of the geometric position of this "situation".

3. Methods

This section will introduce the technical details of the proposed topological world model. The algorithms following are expressed in general terms only, ignoring computational details.

3-1. Pre-Processing

Following the idea of representing situations, consisting of readings from different kinds of sensors in a way that they can be compared in one step, and by employing a simple norm, the sensor-samples have to be preprocessed to form a vector of unified elements. In the current system passive light and tactile sensors as well as an (x, y)-position produced by odometry are available. Considering the fact, that the angular resolution of the tactile sensors is very low, each vector of tactile readings is smoothed by applying gaussian functions. Finally the sensor samples from different types of sensors are weighted and concatenated to produce a "situation-vector", or more briefly a "situation" (consisting of 50 values in the example given).

In the following, sensor situations will be indicated as S; the position or x, y-part of such situations as p.

3-2. Adaptation

As a basis for the network model, the euclidian norm is applied to calculate distances between sensor situations, d, and distances between positions, g, respectively.

Consider a network N consisting of a number of cells c_i, which are connected with respect to the topological neighborhood of the situations $S(c_i)$ attached to each cell. Then at each adaptation step the cell c_{opt} with the smallest situation-distance d_{opt} to the new input situation S_x is determined according to:

$$\forall c_i \in N:$$
$$d_{opt} = d(S(c_{opt}), S_x) \leq d(S(c_i), S_x) \qquad (1)$$

In order to limit the effort for this adaptation to a constant, the search area is limited by the geometric distance g_{search}. In the current system this is done by applying adequate data-structures to the network-

management. The selected cell c_{opt} and all its topological neighbors are then adapted according to:

$$c_{opt}^{new} = c_{opt} - (\varepsilon_o \cdot d_{opt}) \quad (2)$$

$$\forall c_n^j \mid a(c_n^j, c_{opt}) > 0:$$

$$c_{n_j}^{new} = c_{n_j} - (\varepsilon_n \cdot d_{n_j}) \quad (3)$$

where $a(\cdot,\cdot)$ is the adjacency-function of the network. The "classification error" d_{opt} is then added to a total classification error d_{total} attached to the cell c_{opt}.

$\forall c_i \in N$ (after n adaptation steps):

$$d_{total_i} = \sum_{t=1}^{n} hit_{i,t} \cdot d_{opt_t}$$

$$\text{where } hit_{i,t} = \begin{cases} 1 & ; c_i = c_{opt_t} \\ 0 & ; c_i \neq c_{opt_t} \end{cases} \quad (4)$$

In order to decrease the adaptation speed of a well adapted network, the parameters ε_o and ε_n are controlled by:

$$(\varepsilon_o^{new}, \varepsilon_n^{new}) = \begin{cases} (\varepsilon_o \cdot \varepsilon_\Delta, 0) & ; (d_{opt} \leq d_{acc}) \\ (\varepsilon_o^{init}, \varepsilon_n^{init}) & ; (d_{opt} > d_{acc}) \end{cases}$$

$$\text{where: } 0 < \varepsilon_\Delta < 1 \quad (5)$$

and ε_o^{init} and ε_n^{init} are the initial values of the parameters ε_o and ε_n.

In each adaptation step, where d_{opt} is larger than a tolerated error d_{acc}, a global counter n_{miss} is incremented. This counter will be used as a measurement for the need for change in the network structure. An update-counter u_{opt} attached to c_{opt} is incremented and will be used as an indicator for the stability of this specific cell.

In order to use the high speed of this adaptation process to achieve better adaptation, each situation is presented k times to the network. A constant delay of l sensor-sample-time-slots before the current sensor situation affects the network is also found useful (see section "Correlation" below). Accordingly a learning set holding $(k \cdot (l+1)) - 1$ situations is implemented.

3-3. Growing & Shrinking

At start-up time of the system, there are no cells; the network is empty. So the common problem finding a "good" initial state of the network is avoided, but there is a need for some growing strategy. In the present system, two growing strategies are applied. The first is called "**spontaneous insertion**", the second "**statistical insertion**". In the first, new cells, representing the current sensor situation, are inserted when the distance between the current sensor situation and c_{opt} exceeds a certain limit ρ_s (in the special case of an empty network this strategy produces the first cell). In the second strategy a new cell is inserted in the middle between the cell with the highest "degree of movement" c_{runner} (measured by the cell attribute d_{total}) and its farthest topological neighbor c_{far} every n_{insert} "miss-classifications" (measured by the global counter n_{miss}). The new cell is instantiated with mean-values of c_{runner} and c_{far} for position and light-intensity, but with minimal values for touch-information.

Another aspect of growing relates to the topological connections between cells. Assuming that c_{opt} has just changed from c_{opt}^{old} to c_{opt}^{new} in two consecutive adaptation steps, and that the cell c_{opt}^{new} has m other neighbors c_j $(a(c_{opt}^{new}, c_j) > 0)$, the following changes in connection weights are imposed:

$$a(c_{opt}^{new}, c_{opt}^{old}) = 1$$

$$\forall j, (1 \leq j \leq m):$$

$$a^{new}(c_{opt}^{new}, c_j) = a(c_{opt}^{new}, c_j) - (a_{reduce}/m)$$

$$\text{with } 0 < a_{reduce} \leq 1 \quad (6)$$

A connection with a weight ≤ 0 is regarded as non existent. Thus the deletion of cells is now straight forward. A cell or a cell-cluster with no connection is removed.

3-4. Correlation

Three degrees of freedom out of the internal representation ((x, y)-position and orientation) will be corrupted by drift effects or other errors, if they are not continuously correlated to the world model. Even a stable and error-tolerant network structure will not be able to produce stable world models, if the robot's position is corrupted. The obvious approach is to correlate the internal position continuously with the world model build up so far. On the other hand the robot's position is integrated in the world model. This mutual stabilizing technique is very useful when applied to local world models. But, in fact, each representation is updated with the faults, errors and noise from the other. This principal problem prevents a globally consistent world model of arbitrary size including position when only local information is available.

One strategy, used to stabilize the internal position, is to delay the integration of the current sensor situation S_c by a number of adaptation steps l (see section "Adaptation" above). In this way the internal position is correlated on the basis of the formerly integrated world-knowledge at this point, and not on the base of S_c, which would not make any sense. An estimation p_{est} of the current position is produced using the sensor-distances d_i (to S_c) and the number of updates u_i of all cells c_i in the immediate topological neighborhood of S_c. The estimates produced in this way, are not necessarily near to the "correct" position

in every case. Thus the internal position is not changed to p_{est}, but is only "moved" towards this point by a fraction ε_{pos}. The fact that these estimation errors are normally distributed in the long term, together with the fact that the remaining position-errors are integrated into the world model after a certain delay, leads to a stable behavior of the local world model.

The third degree of freedom, the orientation of the robot, is also corrupted by drift and other effects. In order to correlate orientation with the internal representation and the world model, it must be assumed that the current position is correct to within a certain tolerance. Based on the geometric distance, the nearest cell c_{near} is determined by:

$$\forall c_i \in N: \\ g(S(c_{near}), S_x) \leq g(S(c_i), S_x) \qquad (7)$$

To obtain a comparable measurement for the orientation based on sensor impressions including position, we interpret the sensor values as polar coordinates:

$$l_i \Rightarrow (l_i \cdot sin(i \cdot \frac{360°}{s}), l_i \cdot cos(i \cdot \frac{360°}{s})) \\ t_i \Rightarrow (t_i \cdot sin(i \cdot \frac{360°}{s}), t_i \cdot cos(i \cdot \frac{360°}{s})) \qquad (8)$$

where l_i are the light-readings, t_i are the tactile readings and $0 \leq i \leq s-1$. A possible derived measurement for the orientation (modulo 180°) is a linear regression O on the $2s$ polar "sensor-points". The angular difference between the orientation of the regression of the current sensor scan O_c, and the orientation of the regression O_{near} of the nearest cell c_{near}, is interpreted as the current orientation-error. As in the case of the position correlation above, the internal orientation is only "moved" by a fraction ε_{or} towards the estimated orientation. In the current system only the light sensors are employed to determine the orientation estimates. The internal orientation is continuous and not quantised according to the angular sensor-resolution. Thus the incoming sensor-readings are interpolated linearly and rotated according to the current orientation of the robot.

Due to the fact that the remaining position-drift is integrated with the world-model, a global drift of the whole model may occur. This is not necessarily a problem if the world model is drifting as a whole, but as the world model expands and some areas are visited in a sporadic manner, different areas will drift in different ways, resulting in an inconsistent world model. Therefore a cell-specific slow-down of the adaptation speed is introduced when cells are "confirmed" at least n_{fix} times (i.e. $u_{opt} \geq n_{fix}$).

3-5. Results

In the current configuration, 3 sensor-samples per second are employed to build up a stable world model for a mobile robot moving at a speed of 25 cm/s. All computations are completed in realtime on one CPU from the 680x0-class so that the robot does not need to stop in order maintain a consistent world model. Even where each sensor-sample is used 10 times, i.e. 30 adaptations have to be completed each second. Figure 3 shows a network-part of approximately 1 by 1 metre as result of a test-run in an environment measures 3 by 3 metres. The information represented here shows only the position-part of each cell together with their topological relations and the light-distribution associated with.

There is sufficient computational power remaining to run the control-loops of the robot as well as the explorator (see next section) using the same CPU. The navigator is not yet tested under realtime-constraints, but considerations based on the computational effort of the navigator in simulated environments suggests that a realtime-integration using the same CPU seems feasible.

3-6. Evaluation

In this section the features of the qualitative topological world model described above, is compared to the metric modelling technique as applied in the MO-

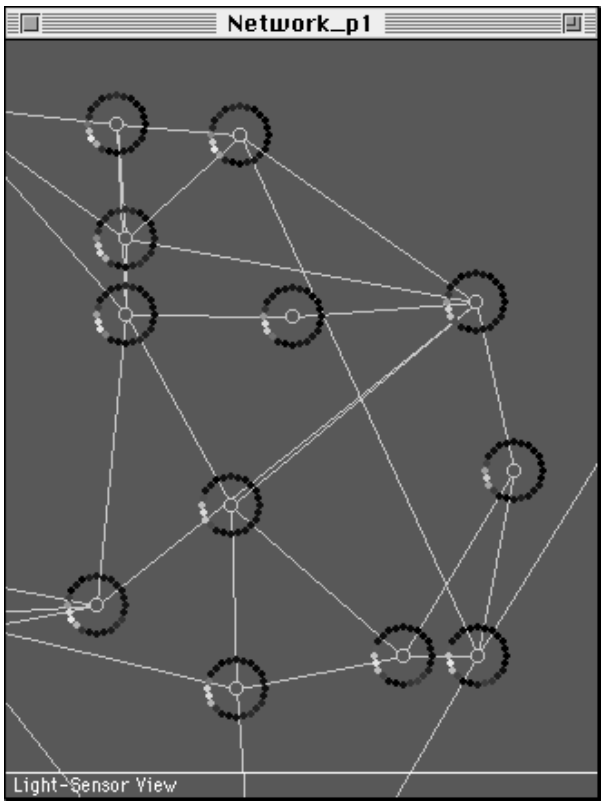

figure 3 : Topological map (light situations)

BOT-project [3], a typical representative from the range of exact geometric modelling techniques.

Computational effort

In the QT-approach one standard CISC-processor (680x0) supplies all the computational power needed under realtime constraints, while multiple CPUs are required for the exact geometric modelling in the MOBOT-project. The amount of energy needed for the CPUs, can be also critical, since the weight of the batteries needed has a significant impact on the construction of the robot-chassis. Moreover the control-effort increases with higher weights, leading to a need for further computational power.

More details about the realtime-abilities of the *ALICE*-project can be found in [15].

Stability

In the exact geometric approach, every sensor-sample has to be evaluated for every disturbance. The only possibilities after such an evaluation are to "accept" or "reject" the sample. Due to the fact that the exact world model is very sensitive to noise, every slightly disturbed sensor-pattern must be rejected, resulting in a very low noise tolerance. On the other hand the QT-model integrates all sensor-samples and smooths the errors by utilizing many sensor-readings in a short time/distance. Assuming that systematic errors are reproducible, they do not lead to system failures in the QT-approach.

Reliability

The benefits of statistical techniques for achieving stability may constitute problems in terms of reliability. A specific situation which occurs only once, for example, finding a useful pathway through a narrow corridor, is ignored by the QT-technique, but not necessarily by the metric modelling process. The second source of problems originates from the technique of using mean-values instead of stored sensor readings in the QT-model. This may lead to inconsistencies, because these mean values may not correspond to real points in the environment. If there is a need for a "totally correct" world model, the metric modelling is called for, although even here nothing is guaranteed. The reliability of the QT-approach can only be expressed in statistical terms. On the other hand the reader should keep in mind, that the real world, or at least the gathered sensor-reading, is inconsistent. Therefore eliminating the inconsistencies in the robot's world model is always a process of approximation. Error-tolerance on the other hand should, in the opinion of the author, be a basic requirement for every useful mobile platform.

Local correlation

The correlation of position- and orientation-drift effects is carried out on a statistical basis in both models (see [13] for a geometric correlation), achieving some compensation for drift effects in both cases.

Sensor Requirements

The differences in the sensor-requirements are obvious. For an exact geometric model we need exact distance measurements, as produced by laser-range finders, radar devices, or large-scale video-signal processing. In order to build QT-Models, any kind of short-range sensor may be considered. In the case of *ALICE*, simple passive photo-resistors together with primitive touch sensors prove sufficient. It should be noticed that most topological models in the literature are built "on top" of a geometric model and therefore use the same sensor-information.

4. Exploration

The proposed topological world model since it is build up continuously and supports livelong learning depends on an "exploration", which has the following three requirements:

- Knowledge should be accumulated **efficiently** in terms of speed, number of rotation or others. This requirement is of course true for all world-model.
- Each sensor-situation has to be stabilized using multiple sensor-samples taken from different positions. To achieve stability, the gathered data must have a certain degree of **redundancy**.
- Due to the need for position- and orientation-**correlation**, the explorator must return the robot regularly to well-established regions of the network.

The last two points limit the amount of exploration-efficiency, which can be expected using the QT-approach.

In the current version, only local information is used to determine a promising exploration-direction. Therefore the current c_{opt} and its immediate neighbors c_j $(1 \le j \le m)$ are used to extract the following features:

- Degrees of cell-confirmation u_{opt} and u_i.
- Number of neighbors of u_{opt}
- Best explored neighbor of u_{opt} with

$$c_{well} \in \{c_j\} \mid \forall j\colon u_{well} \ge u_j \tag{9}$$

- Worst explored neighbor of u_{opt} (analogous to (9))
- Unknown directions leading away from c_{opt} (extracted from the topological structure and memorized as a local attribute at c_{opt})

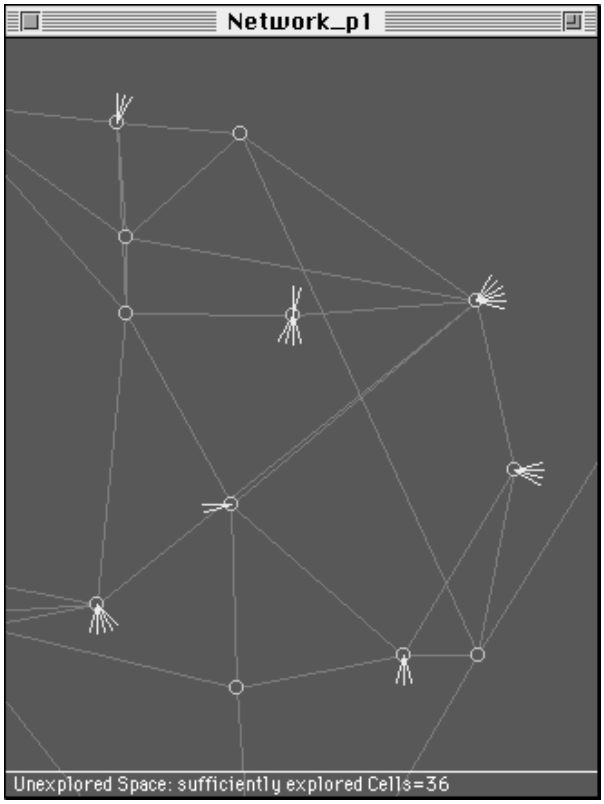

figure 4 : Exploration hints

Based on these features the following "instincts" are evaluated:

- **Stochastic** movements
- **Efficient** movements (smallest number of rotations)
- **Wall-following** (follow an obstacle-border).
- **Network-stabilizing**: If there are more than τ_{sat} neighbors and the current c_{opt} is well explored, drive towards the *least explored neighbor*.
- **Position-stabilizing**: If the current c_{opt} is only less explored, drive towards the *best explored neighbor*.
- **Network-growing**: If there are less than τ_{sat} neighbors try to *create a new neighbor*, by driving towards an unknown direction.

By using a kind of subsumption-architecture, the several instincts are fused to produce a final exploration strategy. In figure 4 the still unexplored directions from the net-fragment shown in section 3 are indicated.

Tests have shown, that an unknown environment of 3 by 3 metres can be explored (i.e. a stable world model, usable for navigation purposes generated) in approximately 15 minutes. Keeping in mind that no range-measuring devices are used in this experiment, this result is (in the opinion of the author) of some significance.

5. Navigation

The navigation module is split into two layers, one concerned with the path-planning task and the other with the problem of translating the planned path into appropriate motor actions.

5-1. Path planning

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

5-2. Adapting motor actions

In order to navigate a planned path, we have to find a mapping from a pair of nodes in the topological map (temporary start and destination points) to appropriate motor commands constrained to a certain accuracy. The kinematics of *ALICE* are quite simple, because we have an omnidirectional platform and high accelerations. The system is able to learn then required mapping even taking account of drift effects, which themselves depend on the actual environment as well as on tolerances in the driving/dead-reckoning part of *ALICE*.

The adaptation of the mapping is done in parallel with the topologic-map-building using a reinforcement learning scheme (figure 5), where supervised dynamic feature maps (introduced in [2]) are applied

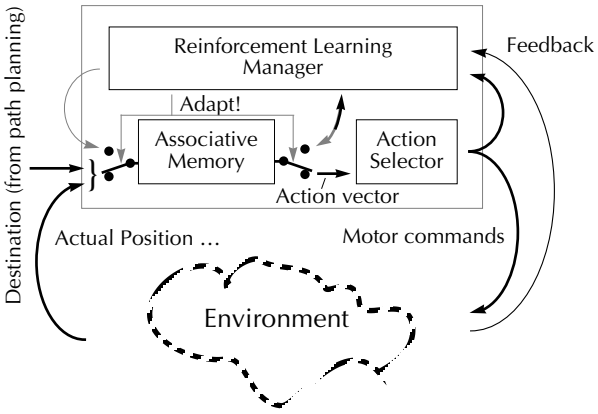

figure 5 : Reinforced learning

in the associative memory module. Later on, the learned translation is applied to areas of the map that were not available during the learning phase (generalisation).

6. Conclusion

As the reader may have expected, the choice of world model will depend significantly on the task, but some guidelines may be derived from the discussion. One aspect is reliability, for example, in a non-error-tolerant environment. If that is to be a central aspect of the robot-task, an exact model may be required to be able to plan safe paths. Similarly, if a guarantee of accuracy when following a path is needed, the exact geometric information may be necessary.

On the other hand, if the main focus is on simplicity, stability or qualitative aspects of the task, the qualitative topological map techniques may be the first choice. In particular, the small requirements for computational effort and sensor equipment together with a high degree of robustness is an unique feature.

References

[1] R. A. Brooks
Solving the Find-Path Problem by Good Representations of Free Space
IEEE Transactions on Systems, Man and Cybernetics, Vol. SMC-13, No. 3, 1983, pp. 190-197

[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] Peter Hoppen, Thomas Knieriemen, Ewald von Puttkamer
Laser-Radar based Mapping and Navigation for an Autonomous Mobile Robot
IEEE International Conference on Robotics and Automation 1990, Cincinnati, Ohio, May 13-18, 1990, pp. 948-953

[4] 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

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

[6] D. Kuan, J.C. Zamiska, R.A. Brooks
Natural Decomposition of Free Space for Path Planning
IEEE International Conference on Robotics and Automation, St. Louis, Missouri, 1985, pp. 168-173

[7] Benjamin J. Kuipers, Yung-Tai Byun
A Robust, Qualitative Method for Robot Spatial Learning
Proceedings of the AAAI 1988

[8] A. Kurz
Building Maps based on the Learned Classification of Ultrasonic Range Data
Intelligent Autonomous Vehicles (IAV) '93, International workshop, University of Southampton, Hampshire, United Kingdom, 18-21 April '93, pp. 193-198

[9] J.-P. Laumond
Model Structuring and Concept Recognition: Two Aspects of Learning for a Mobile Robot
International Joint Conference on Artificial Intelligence, Karlsruhe, '93, pp. 839-841

[10] T. Lozano-Pérez
Spatial Planning: A configuration Space Approach
IEEE Transactions on Computers, Vol. C-32, No. 2, 1983, pp. 108-120

[11] Maja J. Mataric
Integration of Representation Into Goal-Driven Behaviour-Based Robots
IEEE Transactions on Robotics and Automation, Vol. 8, No. 3, June '92

[12] Jun Tani, Naohiro Fukumura
Learning Goal-Directed Sensory-Based Navigation of a Mobile Robot
Neural Networks, Vol. 7, No. 3, pp. 553-563, 1994

[13] Gerhard Weiß, Christopher Wetzler, Ewald von Puttkamer
Keeping Track of Position and Orientation of Moving Indoor Systems by Correlation of Range-Finder Scans
IEEE International Conference on Intelligent Robots and Systems '94 (IROS '94), September 12-16, 1994, Munich, Germany

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

[15] Uwe R. Zimmer & Ewald von Puttkamer
Realtime-learning on an Autonomous Mobile Robot with Neural Networks
Proc. of the IEEE Euromicro '94 Realtime-Workshop, Västerås, Sweden, June 15-17, '94