Robot Navigation Framework Based on Reinforcement Learning for Intelligent Space

László A. Jeni †, Zoltán Istenes ‡, Péter Szemes †, and Hideki Hashimoto †
† Institute of Industrial Science, University of
University, Budapest, Hungary

Abstract — **Navigation in an unknown environment is still a hard problem, because mobile robots need topological maps in order to operate in the environment. Building a map of the environment while also using it for learning is of prime importance for mobile robots but until recently, it has only been confined to small-scale environments.**

This paper describes a mobile robot navigation framework integrated into the Intelligent Space environment. In the Intelligent Space, several Distributed Intelligent Network Devices communicate and share their information about the environment. In this environment mobile robots can be tracked with ultrasonic positioning system and the topological map can be build using laser range finders.

Keywords **— Intelligent Space, object tracking, reinforcement learning, robot navigation.**

I. Introduction

HE field of robotics is closely related to Artificial Intelligence. Intelligence is required for robots to be able to handle such tasks as navigation, referred to as robotic mapping including the sub-problems of localization (knowing where you are), mapping (learning what is around you) and path planning (figuring out how to get there) and such as manipulate objects (usually described in terms of configuration space).

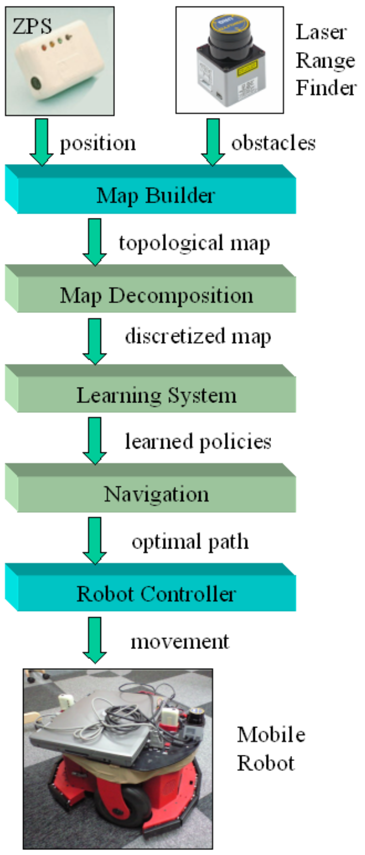
Nowadays mobile robots have multiple functions in a wide spectra of different application areas. The human environment is much more complex and complicated for the robots, they need lots of information about the human environment to be able to achieve their tasks.

There exist a number of researches focused on the intelligent environment. It is worth mentioning here the Interactive Workspace Project [1], in which the possibilities of using ubiquitous embedded sensors and information displays are explored. Another similar research is the Oxygen Project [2]. This research aiming at the development of intelligent environments based on human-centered computation. In the research on Easy Living Technologies [3] human tracking technology is used with the purpose to guess the intent of users in the space in order to automate and facilitate everyday tasks.

In this paper we present an extension of our previous work [4]. This framework uses the capability of the Intelligent Space to build a topological map of the environment and it is able to learn optimal navigation paths using this map.

An overview of the whole system can be seen in Fig. 1. The Intelligent Space can recognize and track the path of mobile robots and we can use this capability to create a topological map of the environment. We can use this feature to make a hierarchical generalization in the learning process by abstraction.

This paper is organized as follows. The next section introduces the Intelligent Space concept. Section 3 briefly describes localization and map building procedure used in the framework. Section 4 describes learning method and the hierarchical decomposition. Section 5 shows experimental results.



II. The Intelligent Space

Conventionally, there is a trend to increase the intelligence of a robot operating in a limited area. The Intelligent Space concept is the opposite of this trend [5]. This is a space where multiple spatially distributed sensors are embedded. The information from the sensors is used by computers and robots connected through a communication network in order to provide various services to humans. The various devices cooperate with each other autonomously, and the whole space has high intelligence based on ubiquitous computing, which is used mainly for welfare support.

A. *Basic Concept*

The Intelligent Space (iSpace) is a space (room or corridor), which has ubiquitous distributed sensory intelligence (various sensors, such as cameras, ultrasound and laser range finders) and actuators (TV projectors, speakers, and mobile agents) to manipulate the space. For the illustration of the main concept of the Intelligent Space see Fig. 2.

A space becomes intelligent when Distributed Intelligent Network Devices (DINDs) are installed in it. A DIND has a sensing function through devices such as a camera and microphone that are networked to process the information in the Intelligent Space. The DINDs monitor the space, achieve data and share them through the network. The iSpace also consists of humans and not only of sensors, cameras or robots. For instance, the iSpace can recognize humans, track their movement to identify the walking areas and learn the shortest safest path in the environment [6]. Subsection B introduces the DINDs partially.

The iSpace is a system for supporting people in it. Events, which happen in it, are understood. However, to support people physically, the intelligent space needs robots to handle real objects. Mobile robots become physical agents of the Intelligent Space and they execute tasks in the physical domain to support people in the space. Moreover, robots can understand the requests (e.g. gestures) from people more effectively. The applicable tasks include movement of objects, providing help, for example to aged or disabled persons etc [7].

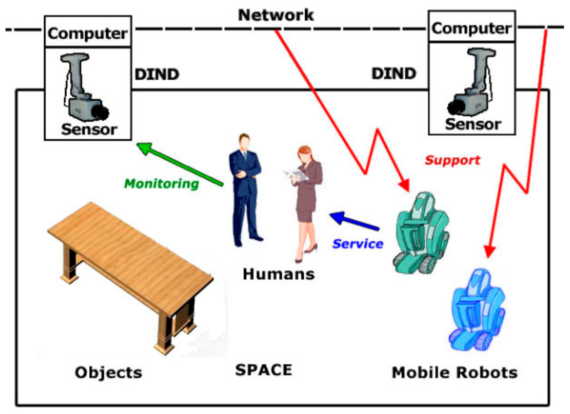


Fig. 2. Intelligent Space Concept.

These mobile robots are called mobile agents. Mobile Agents cooperate with each other and with the other components of the iSpace to realize intelligent services to inhabitants. Robots with their partial intelligence become more intelligent through interaction with the iSpace. Another interesting application here is that the space can serve as a high level and context sensitive interface to robots. The Intelligent Space can physically and mentally support people using robots and by using virtual reality technologies; thereby they can provide satisfaction to people. These functions will be an indispensable technology in the coming intelligence consumption society.

The ongoing research activities about Intelligent Space achieved several results and solutions in the field of motion control [8], feature extraction [9] and recognition and tracking the path of moving objects [10].

Recent research focuses on opto-mechanical contour detection [11], spatial memory [12] and requirement verified mobile code based robot controlling [13].

B. *Distributed Intelligent Network Devices*

Distributed Intelligent Network Devices are the building blocks of the Intelligent Space. The key concept of DIND consists of three basic elements, which are the sensor, the processor (computer) and the communication device (Fig. 3).

Thus, a DIND is a unit based on three functions:

- the dynamic environment (which contains people, vehicles and robots, etc.), is monitored by the sensor,
- the information is processed into a form easily captured by the clients in the processor and
- the DIND communicates with other DINDs through a network.

By installing DINDs into the whole iSpace, they will perform information exchange and information share by communicating mutually through the network. Robots are able to use resources of DINDs as their own parts. However, robots with their own sensors may be considered mobile DINDs. The space thus equipped will became an intelligent ubiquitous computing environment.

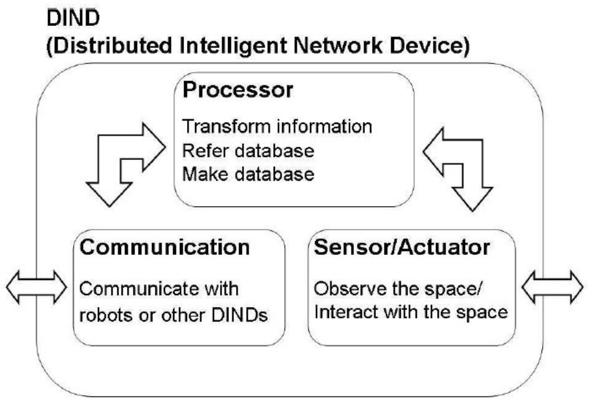

DIND
(Distributed Intelligent Network Device)
Processor
Transform information
Refer database
Make database
Communication
Communicate with robots or other DINDs
Sensor/Actuator
Observe the space/ Interact with the space

Fig. 3. Physical and functional structure of a DIND.

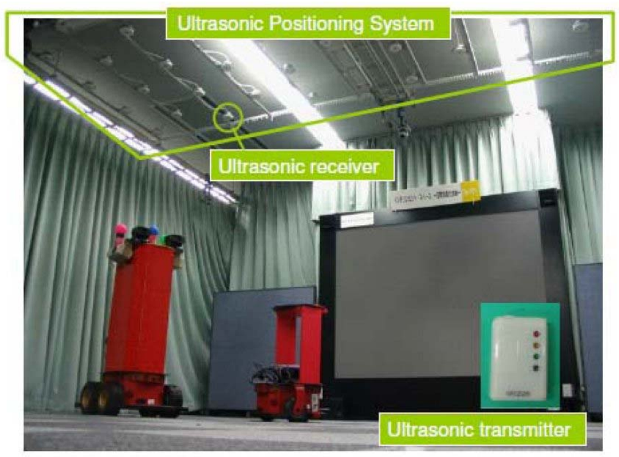


Fig. 4. Zone Positioning System (ZPS).

III. Localization and Map Building

To build the topological map of the environment the system propagates mobile robots in the space and tracks their movement.

A. *Tracking of the mobile robot*

For the mobile robot localization and tracing we used Zone Positioning System [14] by Furukawa Electric, Ltd. (see Fig. 4).

This positioning system is able to measure three-dimensional positions of ultrasonic transmitters. The transmitters are activated based on time division multiple access method. Therefore, sampling frequency depends on the number of transmitters.

The robots are tracked by using two ultrasound tags and by fusing the measurement with readings from the robot's wheel encoders and the sensor fusion is done with a Kalman filter [15]. Using these data we can calculate the exact position and heading direction of the robot (see Fig.5).

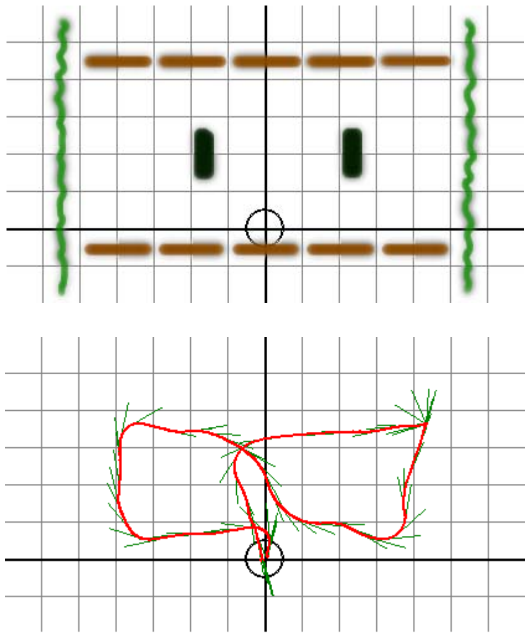


Fig. 5. A simple test scene (up) and the path and heading of the robot in the scene (down).

Fig. 6. Laser Range Finder.

B. *Obstacle Detection and Map Building*

For obstacle detection we used a Hokuyo URG-04LX type laser range finder (see Fig. 6) located on the front side of the robot.

It has a scan angle of 240 degrees with angular resolution of about 0.36 degrees. The measurement error is about 10 mm for the 1 m range, and about 1% for larger distances. The main possible disadvantage for using this device is its maximum measurement distance of only 4 meters. This is a very small range compared to other available laser range finders on the market, but it's much more inexpensive, which makes it a good and affordable option for tracking sensors for indoor Intelligent Spaces and for mobile robot's onboard sensors.

An example of a reading from a laser range finder is shown on Fig. 7. Using the position and heading information provided by the ultrasonic system we can transform the reading from the onboard laser range finder to a global coordinate-system to build a topological map of the space (see Fig. 8).

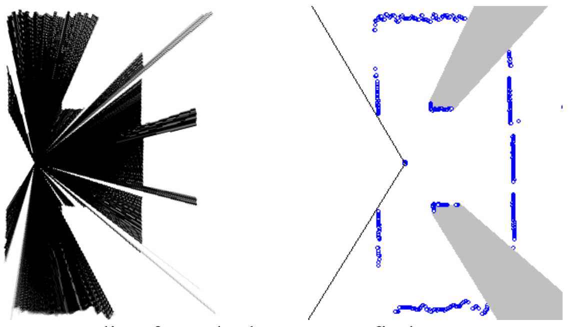


Fig. 7. Reading from the laser range finder.

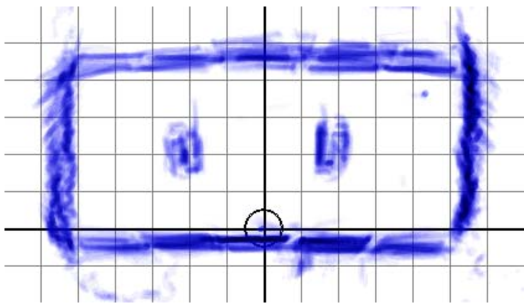


Fig. 8. Topological map of the environment.

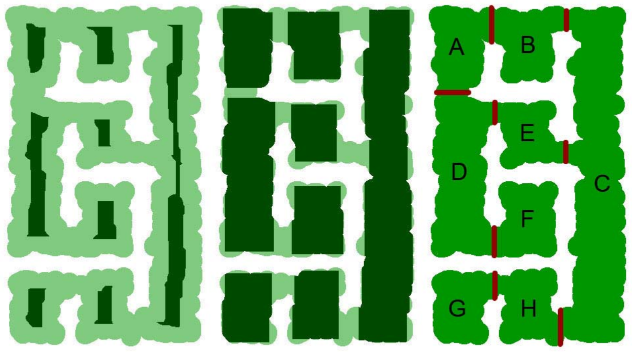


Fig. 9. The eroded (left) and dilatated (center) maps. The final decomposed map can be seen on the right.

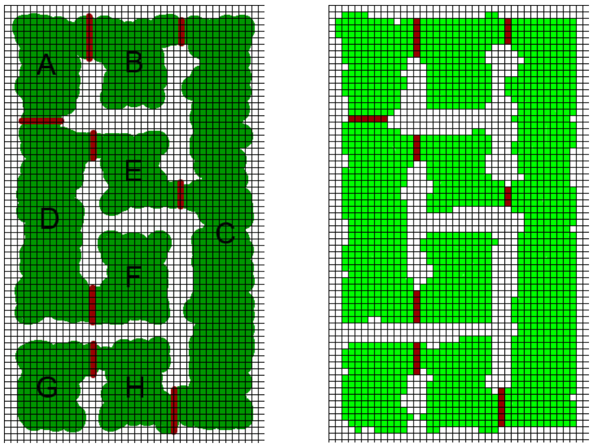


Fig. 10. The discretized map.

C. Map Decomposition

In the next step the system decomposes the whole map into connected sub-maps. This decomposition will be the base of the temporal abstraction in the learning system.

For the decomposition we used two morphological operators. First, we eroded the topological map to break it into partitions, and then a dilatation operator was used to connect these parts again. At the end of this process we can allocate the passages between the partitions. The decomposition steps of a sample map can be seen in Fig.9.

Before the walkable area map is given to the learning system, the system discretizes the map into squares, to reduce computational complexity of the learning. Each square represents a state in the learning problem (see Fig.10).

IV. Reinforcement Learning

A. Basic Concept

The easiest way to describe the concept of a reinforcement learning problem is by considering an agent situated in some environment as shown in Fig. 11.

The agent can detect information about the state of the environment. The agent can also affect the environment by taking one of a set of actions available to it. After each action is taken, the agent receives a feedback signal from the environment called the reward, which determines how well the agent is performing the target task in the environment.

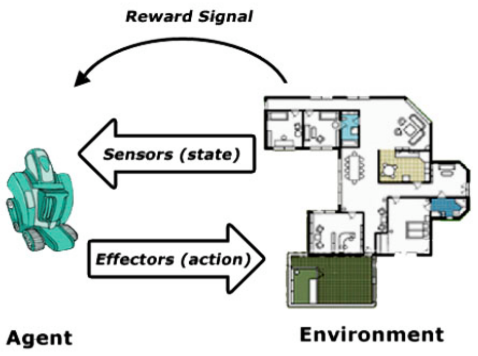


Fig. 11. Basic components of a reinforcement learning problem.

The goal in a reinforcement learning problem is to learn which action to take in each state to maximize some optimality criterion based on the rewards received over time. Some examples of optimality criteria are average reward per time step, total return over a finite horizon and total discounted return.

A reinforcement learning problem can be formalized as a Markov Decision Process or MDP. An MDP is described by a quadruple <*S, A, T, R*> where:

- *S* is the set of possible states.
- *A* is the set of available actions.
- $T(s,a,s') \to [0,1]$ is the transition function defining the probability that taking action *a* in state *s* will result in a transition to state *s'*.
- $R(s,a,s') \to \mathbf{R}$ is the reward function defining the reward received when a transition is made.

A particular strategy for choosing actions in an MDP is known as a policy, and is specified formally as a function $\pi(s,a) \to [0,1]$, which defines the probability of selecting each action in a given state.

For some policy π and a discount factor $\gamma \in [0,1)$, the value function $V^{\pi}(s)$ can be defined as the expected total discounted return when starting in state *s* and using policy π to choose actions:

$$V^{\pi}(s) = \sum_{s'} T(s,\pi(s),s')\left[R(s,\pi(s),s') + \gamma V^{\pi}(s')\right] \quad (1)$$

Intuitively, the value function $V^{\pi}(s)$ represents how good it is for an agent to be in a particular state of the MDP, given that subsequent actions are chosen using policy π.

The discount factor is used to determine the relative worth of future rewards in comparison to rewards available immediately in the current state. The value of γ is chosen to be less than 1 to give V^{π} a finite value for each state. The optimal policy π^* is the policy which, according to the optimality criterion, performs better in the environment than any other policy π. The formal definition of π^* is:

$$V^{\pi^*}(s) = \max_{\pi} V^{\pi}(s), \quad \forall s \in S. \quad (2)$$

While our goal is to find π^*, MDP solution methods are often based on a calculation of the value function for the optimal policy V^{π^*}, also denoted by V^*. Once V^* has been calculated, the parameters of the MDP can be used to calculate π^* as well:

$$\pi^*(s) = \arg\max_a \sum_{s'} T(s,a,s')\left[R(s,a,s') + \gamma V^*(s')\right] \quad (3)$$

B. Hierarchical Decomposition

The main problem, which arises when reinforcement learning is applied to a larger scale problem, is referred to as state space explosion. Without any prior knowledge, reinforcement learning is almost certainly infeasible for state spaces above a certain size.

We can modify the original reinforcement learning problem to use an interconnection function to decompose the large state space. This function represents our prior knowledge about the problem in a formal way. We can describe this modified problem by a triple $<S, A, i>$, where i is the interconnection function.

At first, we describe the interconnection function and show how it can be used to decompose the problem, and then provide a method for learning the decomposed problem.

C. The interconnection function

The interconnection function defines the probability that a given state is a connecting state, which connects two partitions of the problem. It maps the state of the environment to a single number, a probability value. This represents our prior knowledge about the problem.

$$i(s) \rightarrow [0,1],\ s \in S \quad (4)$$

We can say "bottleneck" states in the state space are interconnecting states, because they separate larger partitions.

For example, in a gridworld problem where the world consists rooms and passages, the passage states are interconnecting states, because they connect two rooms (see Fig. 12).

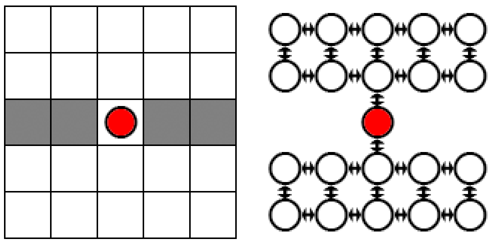


Fig. 12. An interconnecting state in a gridworld. This state connects two rooms

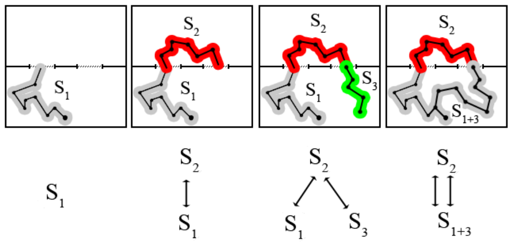


Fig. 13. Four steps from the abstraction process.

If we have some exact knowledge about the problem (for example if we can recognize passages from the current state), we can explicitly define this function. Otherwise we need to approximate it by exploring the state space.

In the framework we used the topological map to define this function.

D. Learning Process using the Interconnection Function

In the first part of the learning process the algorithm explores the state space by starting several trajectories. If we reach an interconnecting state (a state where $i(s) > \delta$, where δ is a constant) we create a new abstract state, which contains the states of the trajectory. If we found a path between two abstract states, which does not contain a connecting state, then we merge the two states. At the end of this stage, we have a set of abstract states and each state represents a partition of the original state space.

Furthermore each abstract state contains one or more connecting states. In the second part of the learning process the algorithm learns partial policies on the partitions represented by the abstract states (see Fig. 13).

Let M denote the number of abstract states. If an abstract state has N_i interconnecting states, then the algorithm learns N_i optimal policies on that partition (one for each connecting state or "passage"):

$$\pi_k^i,\ i \in [1..M],\ k \in [1..N_i] \quad (5)$$

Let A_k^i be a macro action, which represents the shortest path on the ith partition to the kth passage of this partition based on the optimal π_k^i policy. Let S_i denote the ith abstract state. With this notation we can define a Semi-MDP over the abstract states:

$$\left\langle \{\ldots S^i \ldots\}, \{\ldots A_k^i \ldots\}, T', R' \right\rangle \quad (6)$$

We can define the T' transition function using the connectivity information of the partitions.

In the final part of the learning process the system learns the optimal π' policy for this SMDP. This problem has a much smaller state space than the original.

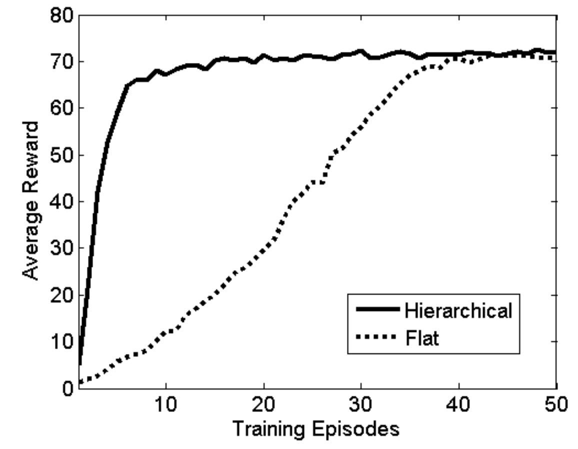
V. Experimental Results

In this section we present some experimental results comparing the flat learning and our hierarchical learning algorithm. Our experiment was performed using a small test scene built in our laboratory.

For the hierarchical algorithm we used the passages on the decomposed map as the interconnection function in the following way:

$$i(s) = \begin{cases} 1, & s \text{ lies on a passage} \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

In the flat problem the agent starts from start position, and gets a reward of +10000 for reaching the goal position. The agent can move in the four directions on the discretized map, with a reward of -1 on every step that does not end at the goal state.



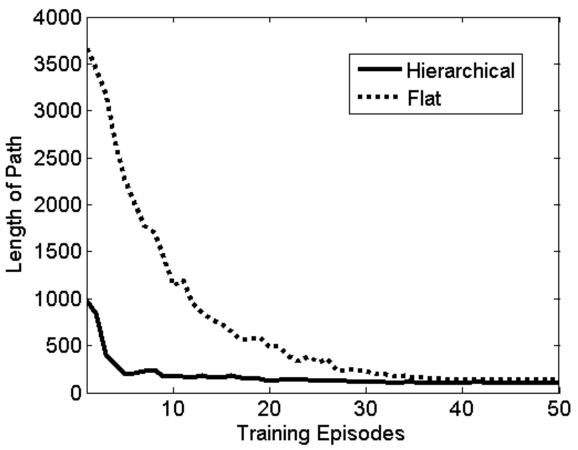


Fig. 14. Average rewards collected during the learning and the length of the optimal path.

In the hierarchical problem the agent starts from a random position on each sub-map, and gets a reward of +100 for reaching a passage on that partition (or the goal position). On the second level of the hierarchy, the agent gets a reward of +100 for reaching the goal state and a reward of -1 on every step that does not end at the goal state.

Fig. 14 displays the learning curves of the flat and hierarchical algorithms obtained by averaging over 100 runs. With the hierarchical decomposition it is possible to learn the task in very few learning episodes. The curve of the flat learner stabilizes around a performance of 72.3 after 38 episodes. The curve of the hierarchical method stabilizes around a performance of 73.1 after 15 episodes.

VI. Conclusion

In this paper we have described a robot navigation framework integrated into the Intelligent Space, which can build a topological map of the environment and uses hierarchical reinforcement learning to reduce the computational complexity of the navigation problem.

Concerning future work, two main open questions could be mentioned. First of all, how to build a reliable topological map in a dynamic environment. In a crowded situation, where the robot is not the only moving entity in the map, the described framework is not able to build the map of the environment.

Secondly, how to create the learning hierarchy without any prior knowledge about the problem. There is still a need for further investigation on approximating the interconnection function by exploring the state space, which will be part of our future work.

References

[1] B. Johanson, A. Fox, T. Winograd, “The interactive workspaces project: Experiences with ubiquitous computing rooms”, IEEE Pervasive Computing Magazine, Vol. 1, No. 2., 2002.

[2] L. Scanlon, “Rethinking the computer – project Oxygen is turning out prototype computer systems”, Technology Review, 2004.

[3] J. Krumm, et al., “Multi-camera person tracking for Easyliving”, in Proc. IEEE Int. Workshop on Visual Surveillance, pp 3-10, Dublin, Ireland, July 2000.

[4] L. A. Jeni, Z. Istenes, P. Korondi, H. Hashimoto, “Mobile Agent Control in Intelligent Space using Reinforcement Learning,” in Proc. 7th International Symposium of Hungarian Researchers on Computational Intelligence (HUCI’06), 2006, pp. 201–210.

[5] P. Korondi, H. Hashimoto, “INTELLIGENT SPACE, AS AN INTEGRATED INTELLIGENT SYSTEM”, Keynote paper of International Conference on Electrical Drives and Power Electronics, Proceedings, 2003, pp. 24-31.

[6] P. T. Szemes, H. Hashimoto, “Estimation of Walking Habit in iSpace”, Proceeding of the 4th International Symposium on Advanced Intelligent Systems (ISIS 2003), pp.531-534, 2003.09, Jeju, Korea, ISSN 1738-0073

[7] P. T. Szemes, J. Lee, H. Hashimoto, P. Korondi, “Guiding and Communication Assistant for Disabled in Intelligent Urban Environment”, IEEE/ASME International Conference on Advanced Intelligent Mechtronics Proceedings (AIM), pp.598-603, 2003.07, Kobe, Japan

[8] P. T. Szemes, “Human Observation-based Motion Control Strategies in Intelligent Space”, PhD Thesis, Tokyo, 2005.

[9] K. Morioka, J. Lee, Y. Kuroda, H. Hashimoto, "Hybrid Tracking Based on Color Histogram for Intelligent Space", Artificial Life and Robotics, Vol.11, No.2, pp.204-210, 2007.07, 1433-5298

[10] Z. Petres, P. Baranyi, P. Korondi, H. Hashimoto, "Trajectory Tracking by TP Model Transformation: Case Study of a Benchmark Problem", IEEE Transacions on Industrial Electronics, Vol.54, No.3, pp.1654-1663, 2007.06, 0278-0046

[11] B. Resko, P. Korondi, H. Hashimoto, "Opto-Mechanical Filtering Applied for Orientation and Length Selective Contour Detection", Proceedings of the 33th Annual Conference of the IEEE Industrial Electronics Society (IECON '07), pp.2553-2558, 2007.11, Taipei, Taiwan, ISBN 1-4244-0783-4

[12] M. Niitsuma, H. Hashimoto, "A Description of Human Activities using Spatial Memory in Intelligent Space", Proceedings of the 33th Annual Conference of the IEEE Industrial Electronics Society (IECON '07), pp.17-21, 2007.11, Taipei, Taiwan, ISBN 1-4244-0783-4

[13] L. A. Jeni, Z. Istenes, M. Tejfel, “Safe Mobile Code in the Intelligent Space”, Symposium of Young Researchers on Intelligent Systems (Intelligens Rendszerek Fiatal Kutatók Szimpóziuma, IRFIX), 2007, Budapest, Hungary.

[14] Zone Positioning System, Furukawa Industrial Machinery Systems CO., LTD, http://www.furukawakk.jp/products/

[15] D. Brscic, H. Hashimoto, “Tracking of Objects in Intelligent Space Using Laser Range Finders”, in Proceedings of the IEEE International Conference on Industrial Technology ICIT 2006, pp. 1723-1728, Mumbai India, December 2006.