

# Rule-Based Adaptive Navigation for an Intelligent Educational Mobile Robot

Mihaela M. Oprea

University Petroleum-Gas of Ploiesti, Department of Informatics,  
Bd. Bucuresti nr. 39, Ploiesti, 100680, Romania, [mihaela@upg-ploiesti.ro](mailto:mihaela@upg-ploiesti.ro)

**Abstract.** The paper presents a hybrid adaptation method that combines a knowledge-based approach with reinforcement learning and a simulated annealing technique, and is applied in the navigation of an educational mobile robot. The experimental results of simulations showed a good behaviour of the robot when doing an adaptive navigation in a dynamic environment by using the proposed hybrid method.

## 1 Introduction

The improvement of a navigation system performances can be achieved by using different artificial intelligence (AI) techniques such as knowledge-based approaches [11], adaptation methods [2], genetic algorithms, genetic programming [1] and so on. Moreover, the navigation task in a dynamic and uncertain world is a key challenge for mobile robotics [5], [15]. In this paper it is proposed a hybrid adaptation method for the navigation of an educational mobile robot that works in a dynamic environment. The hybrid adaptation method combines a knowledge-based approach with a reinforcement learning technique (RL) and a simulated annealing technique (SA). Our research work involves the application of such a hybrid method in educational mobile robotics for two main types of navigation experiments: simulations and real-world experiments that use a toy mobile robot, ROBO, developed at University of Ploiesti. In our experiments we have considered indoor environments such as offices and homes.

The paper is organized as follows. Section 2 describes the robot navigation task. The intelligent educational mobile robot system is presented in section 3. The architecture of the system as well as the ontology with its associated knowledge base, and adaptation method are detailed. Some experiments and preliminary results are discussed in section 4. The last section concludes the paper and highlights some future work.

---

Please use the following format when citing this chapter:

Oprea, Mihaela, 2006, in IFIP International Federation for Information Processing, Volume 204, Artificial Intelligence Applications and Innovations, eds. Maglogiannis, I., Karpouzis, K., Bramer, M., (Boston: Springer), pp. 35–43

## **2 Robot navigation task**

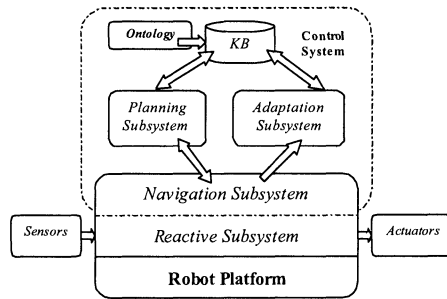
The navigation task has to direct the robot to a given location, avoiding obstacles. Its input is a path plan. Basically, navigation is the process of finding clear paths, avoiding collisions with obstacles and calculating the robot's current velocity and orientation. A plan for the navigation problem is a path through the world from the initial state (start state) to the goal state (final state). Usually, the initial state include the complete map of the world. During navigation some unexpected situations may arise, and the robot cannot follow the plan, so it needs to replan starting from the current position as the new start position. In real-time systems it is necessary to tradeoff the quality of a plan against the computational cost of planning, and it is necessary to interleave planning and execution, to combine reactive with deliberative capabilities. Several classes of planning algorithms are used in robot navigation [11]: cell decomposition methods, skeletonization methods, online algorithms, bounded-error planning methods, landmark-based navigation methods.

## **3 The intelligent educational mobile robot system**

For academic and research purposes we have developed an intelligent educational mobile robot system, ROBO, that has a virtual implementation for simulations and a physical implementation for real-world experiments. The robot is used as a testbed for the study of different AI and machine learning techniques taught at the courses of Artificial Intelligence and Intelligent Agents to different categories of students, undergraduate, masterate, and postgraduate. In this section we shall focus on the architecture of the system, on the domain knowledge base and its associated ontology, and on the adaptation method.

### **3.1 The architecture**

The architecture of the intelligent educational mobile robot is composed by the following subsystems: the reactive subsystem (RS), the navigation subsystem (NS), the planning subsystem (PS), the adaptation subsystem (AS), and a knowledge base (KB). Figure 1 shows this architecture.



**Fig. 1.** The architecture of the intelligent educational mobile robot.

The reactive subsystem performs collision avoidance and in collaboration with NS path following. It consists of several processes for perception and action. For example, RS will include the basic four actions (move forward, turn left, turn right, move backward) and sensors related processes. Basically, RS contains a set of behaviours for coordinating perception and action. The navigation subsystem, which works together with RS, performs the robot positioning and path following and has to deal with sensor and actuator uncertainty. The control system is composed by the planning subsystem, the adaptation subsystem and the knowledge base, including also the ontology used. The planning subsystem computes the optimal path that connect the starting position and the destination position. In the planning process several rules from KB are used. The adaptation subsystem will help the robot when unexpected obstacles occur, suggesting solutions to various situations. The main adaptive method used is reinforcement learning [14]. Also, a simulated annealing technique is used in order to help the robot moving out from deadlock locations. The knowledge base contains facts and rules that could be used in both planning, and adaptation processes during the mobile robot navigation.

### 3.2 The ontology

Built on a particular domain of knowledge, an ontology [4] contains the terminological primitives of the domain structured in a set of concepts (terms) which represent the objects of the domain, and a set of relations between these concepts. All the concepts related to a mobile robot navigation (i.e. directly related to the robot and its environment) were grouped in an ontology named *OntoRobotic*. Each term is defined, characterized by its properties, constraints, and classified in a hierarchy of classes that form the ontology tree. Figure 2 shows a part from the ontology tree. Example of terms used are: ROBOT, SENSOR, ACTUATOR, INFRARED, ODOMETER, WHEEL, ENVIRONMENT, BEHAVIOUR, DOOR, OBSTACLE AVOIDANCE. Some relations between different concepts are: close-to, next-to, closed, open, front, back, left, right etc.

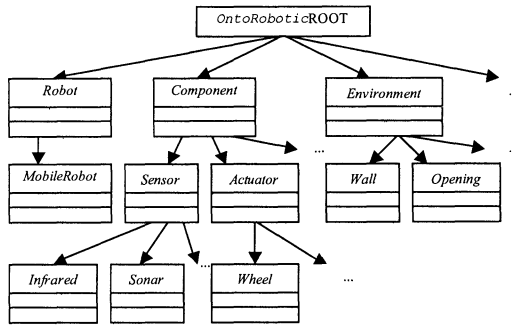


Fig. 2. The ontology tree (selection).

The ontology `OntoRobotic` was developed in Protégé [10], an ontology editing environment based on Java. Starting from the ontology it is generated a knowledge base specific to the mobile robot navigation tasks.

### 3.3 The knowledge base

The knowledge base contains the knowledge that is used in the navigation task. The knowledge representation method used is the production rule form, i.e. *if-then* rules. KB includes rules generated from the `OntoRobotic` ontology as well as rules that are directly connected with typical situations that may arise during robot navigation.

*Examples of rules:*

Situation: *moving toward object O*

**Rule NTO15** (Robo next to the object O)

**if**  $distance(Robo, O) < Th^{NT}$  **then**

\* move back with the distance  $\Delta d$

**call** `obstacle_avoidance`;

**Rule CTO16** (Robo close to the object O)

**if**  $distance(Robo, O) < Th^{CT}$  **then**

**call** `obstacle_avoidance`;

where the relation between the two thresholds,  $Th^{NT}$  and  $Th^{CT}$  is  $Th^{NT} > Th^{CT}$ .

Situation: *crossing doors*

**Rule CD5**

**if** `door(D2, Open)` **then**

**if**  $Prob(D_1, Closed) < Th^D$  **then**

**return** `Path_B`;

**return** `Path_A`;

where  $Prob(D=Closed)$  represents the probability that door  $D$  is closed and  $Th^D$  is a threshold that is chosen by taking into account the initial values for the probabilities regarding the state of doors that could be crossed by the robot.

Situation: *obstacle avoidance*

**Rule OA10**

**if** `RoboState(Robo, obstacle_avoidance)` **then**

\* follow the obstacle and keep moving toward the goal or try to turn left or right

**if** `RoboState(Robo, blocked)` **then**

\* apply a simulated annealing technique (SA), i.e. move the robot in a certain direction (chosen as given by SA) in order to move out from the deadlock;

The knowledge base includes also a group of rules related to different adaptation situations. In the next subsection we shall focus on the reinforcement learning technique that is used by the educational mobile robot system.

### 3.4 The adaptation method

The successful applications of reinforcement learning technique to navigation tasks (see e.g. [3], [12], [16]) has determined us to use as adaptation method a reinforcement learning technique. The basic concept behind reinforcement learning technique is that if an action is followed by a satisfactory response, then the tendency to produce that action is strengthened, i.e. reinforced. The learning system receives state information about the environment by means of its sensors, and this state information is used by a reasoning process to determine the action to be taken in the given state. The goal of the learning task is to associate with each state the best action to be selected such that the reward from the environment is maximized [7].

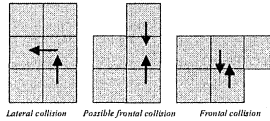
The combination of the navigation and learning methods reported in [8] and [9] had generated the hybrid adaptation method that is used by our educational mobile robot. The reinforcement learning technique was adapted to ROBO and it is briefly presented. Let's consider that a mobile robot RoboS has to find the optimal path in a grid world. The actions set  $A$  is composed by the following operators: *move\_north*, *move\_south*, *move\_est*, *move\_west*, *move\_NE*, *move\_NW*, *move\_SE*, *move\_SW*, *hold*. The action *hold* means that the robot will stay at the current location. The goal of RoboS is to learn moves that quickly takes it to its respective goal destination without colliding with the obstacles (static or dynamic) that are or arise in the environment. As the robot knows its current position and the goal position, it can recognize if it moves to or away from the destination. The feedback comes from the robot's sensors via the embedded adaptation subsystem. Thus, the robot  $R$  receives a feedback,  $feedback(R)$ , based on its movement as given by relation (1).

$$feedback(R) = \begin{cases} 1, & \text{move toward R's goal} \\ -1, & \text{move away from R's goal} \\ 0, & \text{no change} \\ fc(R, O), & \text{a collision arise} \end{cases} \quad (1)$$

The feedback is 1 in case the robot moves toward its goal. If it moves away from its goal, the feedback is -1. In case there is no change of the distance between the robot and the goal, the feedback is 0. If a collision arise between the robot  $R$  and an obstacle  $O$ , then it is used the collision function,  $fc(R, O)$  given by relation (2).

$$fc(R, O) = \begin{cases} -4, & \text{lateral collision} \\ -4, & \text{possible frontal collision} \\ -8, & \text{frontal collision} \end{cases} \quad (2)$$

When the robot is located next to an obstacle, it is very likely to produce a collision. Figure 3 shows the three types of possible collisions.



**Fig. 3.** The possible collisions.

## 4 Preliminary experimental results

The current research work involves two types of experiments, simulations, and real world experiments in indoor environments with the educational robot system, ROBO, developed in the Department of Informatics, University of Ploiesti. The toy mobile robot ROBO (presented in Figure 4) has three sensors, two collision sensors, and one IR sensor. So far, it was studied the application of the hybrid adaptation method in a simulated environment in which a simulated mobile robot is doing navigation tasks. In this section we shall present the simulation results.

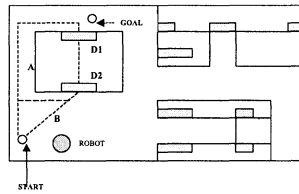


**Fig. 4.** The educational toy robot ROBO.

*Experiment 1* (no map use *versus* map use in the navigation task):

The first set of experiments considered an analysis of behaviour-based navigation (with no map of the environment) and an A\*-like navigation (with a topological map of the environment). Let's consider the navigation scenario presented in Figure 5. We assume that the mobile robot has 16 sonar sensors arranged in a ring and the following set of possible actions are allowed:

*wall\_following* (go\_forward, go\_backward), *turn\_left* ( $\pm 90^\circ$ ,  $\pm 45^\circ$ ), *turn\_right* ( $\pm 90^\circ$ ,  $\pm 45^\circ$ ), *avoid\_obstacle*

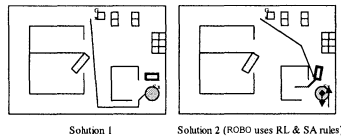


**Fig. 5.** Path planning – a door is closed.

Two existing planning methods are analysed on different situations that simulate a dynamic and uncertain world in which the robot has to do its mission. Both methods use sensor model (actually, sensor fusion) that include the possibility of failure. The first method A will make a behaviour-based planning that has no map of the world, but instead it has a knowledge base that include *if-then* rules (with uncertainty factors) that characterize different behaviours, given a policy to the planner. The second method B, similar with that described in [13], uses a topological map and an A\* algorithm to generate several paths that will be used in conjunction with some behaviours by the replanning algorithm. A complete set of behaviours is developed, considering all the possible situations that may arise (including the worst cases). In order to reduce the complexity (i.e. the number of behaviours) we have made a partition of the possible situations. Figure 5 shows the experimental results obtained for the case in which door  $D_2$  is closed. Method A will give the solution Path A (the best solution in that particular case), while method B will give path B, much longer than path A. In this case method A worked better than method B. A possible solution is to extend the behaviour set with a number of negative behaviours [16] that will be used by the planning/replanning algorithm of method B. Critical situations such as closing a door, blocking at an obstacle avoidance are managed by special rules that exists in the knowledge base of the navigation system. These rules will lead to better solutions. The method will give the safest or the shortest path according with the needs of the navigator task. This specification, shortest or safest path, will be used to choose the admissible heuristic function for the A\* algorithm. The modified version of method B gave a reliable and robust navigation in the case of a simulation of the navigation task with a mobile robot that moves around in a simulated dynamic and uncertain environment.

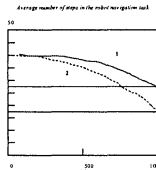
#### *Experiment 2* (the use of reinforcement learning):

Let's consider the scenario from Figure 6, where a moving object will lock the first attempt of the robot movement, that of going in the north direction. Two solutions could be adopted. In case of the second solution (use of RL&SA), as the obstacle will move into the room, it will clear the way of the robot to its destination, and the robot will take the best decision. The path will be determined by the following sequence of operators  $\{move\_south, hold, hold, move\_north, \dots\}$ , while simultaneously, the moving obstacle will do the sequence of actions  $\{move\_SW, move\_west, hold\}$ . Therefore, the lessons learned during adaptation (i.e. to hold one or two times) help the robot to have a better behaviour, that of choosing solution 2.



**Fig. 6.** Example of a navigation scenario – two solutions.

During experiment 2 we have analysed the Q-learning performance in terms of the average number of steps followed in the robot navigation task. The experimental results obtained are averaged over 50 runs. In order to attain convergence we have set the number of trials at 1000. As shown in Figure 7 the average number of steps is acceptable. Graphic 1 shows the average number of steps in the robot navigation task, in case the system parameters setting is  $\beta=0.3$ ,  $\gamma=0.5$ , while graphic 2 corresponds to the system parameters  $\beta=0.4$ ,  $\gamma=0.7$ .



**Fig. 7.** The performance of Q-learning.

## 5 Conclusion and future work

The inclusion of an adaptation capability in the architecture of a mobile robot can improve the robot navigation tasks in terms of time and path length. In this paper we have presented a hybrid adaptation method that combines a knowledge based approach with a reinforcement learning technique and a simulated annealing technique. The adaptation method was applied to an intelligent educational mobile robot. The control system of the mobile robot has a knowledge base that includes knowledge under the form of facts and production rules that specify the robot behaviour in certain situations such as obstacle avoidance, crossing doors, U-shaped traps. Basically, the robot guidance during navigation is realized by the rules from the knowledge base that apply also simulated annealing techniques and/or reinforcement learned behaviour rules by taking into account the current state of the mobile robot navigation. The simulations made so far showed a good performance of the hybrid adaptation method in a dynamic and uncertain world. As a future work, we shall analyse the use of the hybrid adaptation method in a real-world indoor environment navigation made by the educational toy mobile robot, ROBO.



## References

1. Dain, R. A.: Developing Mobile Robot Wall-Following Algorithms Using Genetic Programming. *HTR Labs. Technical Report* (1998).
2. Fukuda, T., Kubota, N.: Learning, Adaptation and Evolution of Intelligent Robotic System, *Proceedings of the IEEE Int. Symposium on Intelligent Control*, Maryland, USA (1998) 2-7.
3. Goldberg, D., Mataric, M.: Reward Maximization in a Non-Stationary Mobile Robot Environment. *Proc. of the 4<sup>th</sup> Int. Conf. on Autonomous Agents*. Barcelona (2000) 92-99.
4. Gruber, T. R.: A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2) (1993) 199-220.
5. Iocchi, L., Lukasiewicz, T., Nardi, D., Rosati, R.: Reasoning about Actions with Sensing under Qualitative and Probabilistic Uncertainty. *Proceedings of the 16<sup>th</sup> European Conference on Artificial Intelligence*. Valencia, Spain (2004) 818-822.
6. Latombe, J.-C.: *Robot Motion Planning*. Kluwer Academic Publishers, Boston (1991).
7. Mitchell, T.: *Machine Learning*. McGraw-Hill. Boston (1997).
8. Oprea, M.: Path Planning in a Robot Navigation System. *Proceedings of 16th IJCAI99 Workshop PLAN-2 Scheduling and Planning meet Real-time Monitoring in a Dynamic and Uncertain World*. Stockholm. (1999) 65-70.
9. Oprea, M.: Reinforcement Learning Applied in Mobile Robot Path Planning. *Proceedings of the 15th Int. Conf. on Control Systems and Computer Science CSCS15*. Bucharest. Romania. Politehnica Press (2005) 447-451.
10. Protégé: <http://protégé.stanford.edu> (2000).
11. Russell, S., Norvig, P.: *Artificial Intelligence - A Modern Approach*, Prentice Hall, New Jersey (1995).
12. Santana, H., Corruble, V., Ratitch, B.: Multi-Agent Patrolling with Reinforcement Learning. *Proceedings of the 3<sup>rd</sup> Int. Joint Conf. On Autonomous Agents & Multi Agent Systems*. ACM Press. Vol. 3. New York, USA (2004) 1122-1129.
13. Simmons, R., Goodwin, R., Haigh, K. Z., Koenig, S., O'Sullivan, J.: A Layered Architecture for Office Delivery Robots. *Proceedings of Autonomous Agents '97*, Marina Del Rey, USA (1997) 245-252.
14. Sutton, R., Barto, A.: *Reinforcement Learning: An Introduction*. Cambridge USA (1998).
15. Wolf, D, Sukhatme, S.: Online Simultaneous Localization and Mapping in Dynamic Environments. *Proceed. of the IEEE Int. Conf. Robotics and Automation*. (2004) 1301-1307.
16. Yamaguchi, T., Masubuchi, M., Tanaka, Y., Yachida, M.: Reinforcement Learning for a Real Robot in a Real Environment. *Proc. of ECAI'96*, Budapest, Hungary (1996) 694-698.