A CBR strategy for autonomous reactive navigation learning

E.J. Pérez, C. Urdiales and F. Sandoval, Dpto. Tecnología Electrónica, E.T.S.I. Telecomunicación, Campus de Teatinos, Univ. of Málaga, 29071 Málaga (Spain), cristina,edu,sandoval@dte.uma.es
J. Vazquez, Institute of Information and Computing Sciences, Utrecht University, Utretch (Holland), javier@cs.uu.nl

ABSTRACT

This paper evaluates the learning capabilities of a new Case Based Reasoning based scheme for pure reactive navigation. The system only relies on sensor readings and goal direction at a given time instant. It acquires knowledge about how other systems, human or algorithms, handle reactive behaviors and also learns from its own experience. It has been successfully tested how the system learns from different sources and how it adapts to different situations.

KEYWORDS: Case Based Reasoning Reactive Navigation Learning .

1. INTRODUCTION

Autonomous robotic navigation has been widely studied in the last decades. Much research has focused on navigation in dynamic environments, where changes are difficult to model and predict. When no information is available about an environment, navigation becomes mostly reactive. One of the best known reactive schemes is the potential fields method. Potential fields are simple and efficient, but they also present some drawbacks [5]: i) it is difficult to move between close obstacles (i.e. doors); ii) they tend to present oscillations when the robot is close to obstacles; iii) they may converge to local minima. Thus, sometimes they can not move safely through packed areas. Some approaches to resolve these problems have also been proposed: the Vector Field Histogram [11], the Elastic Bands method [9] or the Dynamic Window Approach [3]. The main drawback of all these approaches is that they depend on many parameters that are difficult to optimize for all cases. Consequently, algorithms capable of learning are more efficient in dynamic environments, instead of providing a single analytical solution to all problems.

Case-Based Reasoning (CBR) is a learning and adaptation technique to solve current problems by retrieving and adapting past experiences [1]: CBR systems become more efficient by remembering old solutions for similar problems and adapting them to new ones rather than solving them from scratch. Each new problem together with its new solution becomes a new case that can also be stored and used later. A CBR cycle consists of the following steps: i) retrieve the most similar stored case; ii) adapt the retrieved case information to fit the current case; iii) evaluate results for feedback about the suitability of the solution; iv) learn from the new experience to use it for future problem solving.

CBR has been used in robotic autonomous navigation schemes before, usually for global path planning in static environments [2], global planning in a priori known dynamic environments [4][7] and non pure reactive navigation [8][10] relying on accumulating experience over a time window. However, no CBR based method has been used for pure reactive navigation where decisions are taken regarding information available only at a given time instant. The authors proposed in [12] such an strategy to teach the robot how humans react given a punctual situation along a trajectory. The input data for the pure reactive scheme was only the sensor readings at a given time instant, the direction to the goal and the current heading of the robot. The target was, obviously, to reach the goal but also to achieve a short, safe and smooth trajectory.

It was stated in [12] that the system worked better if some knowledge was injected to the system a priori, in that case by manually driving the robot through some routes to seed the casebase with some cases. However, it was evaluated how the system performed in a real environment and not evaluated how it progressively learnt. In this paper, the learning process in our CBR reactive system, briefly described in section 2, is tested and evaluated in section 3. Section 4 presents some results and conclusions.

2. A CBR STRATEGY FOR AUTONOMOUS NAVIGATION

In a pure reactive scheme, a case should be defined basically by the sensor readings and the goal. However, to avoid storing too many cases derived from small perception differences, we discretize sonar readings into 5 intervals: i) critical (0-20 cm); ii) near (20-50 cm); iii) medium (50-100 cm); iv) far (100-150 cm); and v) no influence (150-800 cm). Using this information, the system must return a new heading direction for the robot to safely reach the goal in that particular situation. In order to adapt to dynamic environments, the solution to a case can not be judged good or bad in terms of arriving to a goal because most trajectories involve several cases, good or bad. Thus, we evaluate solutions by using three simple local criteria. In order to provide some temporal inertia against sonar noise and to prevent slippage as much as possible, we try to minimize curvature changes in the robot trajectory. The second evaluation factor is the angle between the robot-goal vector and the heading direction, so that the robot avoids getting too far from its partial goal. Finally, to avoid getting too close to obstacles, the smaller sonar reading is also included as an evaluation factor. Fig. 1 shows how a case is defined for a robot equipped with 5 sensors and heading to the direction marked by the black arrow. The case includes the sensors readings plus the goal direction, marked with a gray arrow. The evaluated criteria are the shortest distance to obstacle d_{min}, the curvature variation $\alpha 1$ and the angle formed by the heading direction and the goal direction $\alpha 2$.

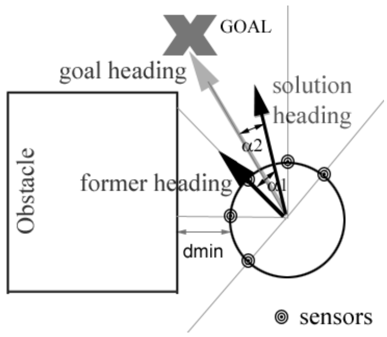

Fig. 1A case parameters and evaluation factors.

Using the CBR scheme proposed in [12], the robot tries to reach goals in a straight way but as soon as obstacles are detected nearby, the most similar case in the casebase to the current situation is retrieved and the robot heading changes accordingly. However, if the retrieved case is too different from the current situation, the solution may mutate by correcting the returned heading with repulsors from the closest obstacles detected. These mutated cases can also be stored to help in future situations. Several cases concatenated plus a goal-attraction behavior in absence of obstacles nearby finally make the robot reach the goal.

3. REACTIVE LEARNING FOR AN AUTONOMOUS ROBOT

According to the proposed scheme, a robot can either learn from scratch, from an off-line supervised training stage or from its own experience. In this section, we present several experiments to evaluate the impact of all learning strategies. All experiments have been performed with a simulator for two main reasons: i) to keep exactly the same training conditions in all cases for the sake of comparison; and ii) to avoid sonar effects so that it can be clearly understood how the system behaves in absence of sonar errors. All tests have been performed with one, two and three obstacles in the way of the robot.

3.1 Off-line learning from a human driver

Our first set of experiments consisted of manually driving the robot around a single obstacle through four different routes (Fig. 2). It can be observed that some of these routes were purposefully jerky to test if the robot was actually capable of choosing the best option to navigate on its own after training was complete.

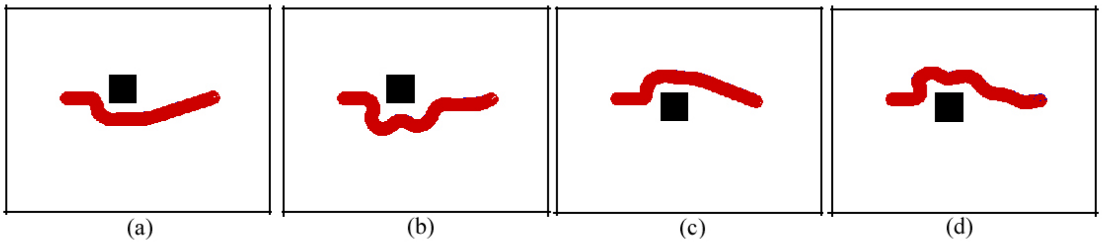

Fig. 2: a-d) Trajectories guided by a human around a single obstacle.

Fig. 3.a shows how the robot perform on its own in a very similar situation after only the trajectories in Figs. 2.a-b have been learnt by the robot. White circles over the trajectory show where the CBR reactive layer is triggered and black ones show where cases are mutated. It can be observed that the robot tries to avoid jerky behavior, even though purposefully included in training. It can be appreciated that even for this known situation, the robot decides that some cases are not good, specifically because we purposefully drove the robot close to the obstacle in the learning stage. Thus, it can be observed that cases where the robot is too close to obstacles are mutated. Nevertheless, we did not include these mutated cases in the CBR yet to test how the robot performs in more complex situations when its knowledge is still poor.

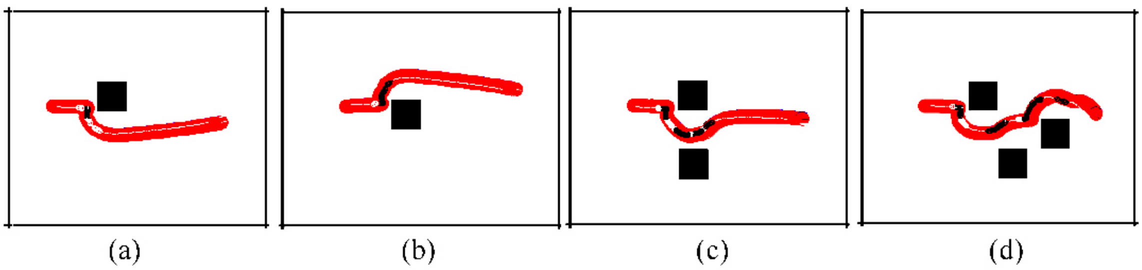

Fig. 3:robot trajectory after training in Figs. 2.a and b with: a) obstacle on the left; b) obstacle on the right; c) obstacles at both sides; d) three obstacles in the way.

Fig. 3.b shows the robot performance after training in Figs.2.a-b when there is also a single obstacle in the way. However, it must be noted that this time the obstacle is left on the right side of the robot, because leaving it on the left would mean a much longer trajectory. The interesting point about this is that the robot never learnt how to leave an obstacle on the right. Nevertheless, the CBR promptly adapts to the new situation by mutating most cases where the robot is turning. Again, we do not include these cases for tests in Figs. 3.c-d, where we can observe how the robot deals with two and three obstacles respectively. As expected, situations where the robot finds obstacles on both sides or on the right, cases are mutated. Despite the need for mutations, it can be observed that the resulting trajectories are mildly smooth, short and safe. However, in order to test how good these trajectories are, it is better to compare them with those returned by a well known method in the same conditions. We have implemented and adjusted a

standard potential fields method and its results for the same four situations are presented in Figs.4.a-d. It can be observed that the resulting trajectories are mostly smooth and short, even though they tend to present oscillations, as expected. The experiments in Figs.3.a-d had less oscillations, but since the robot had learnt to move close to obstacles, they still presented sharp trajectory changes when obstacles were very close.

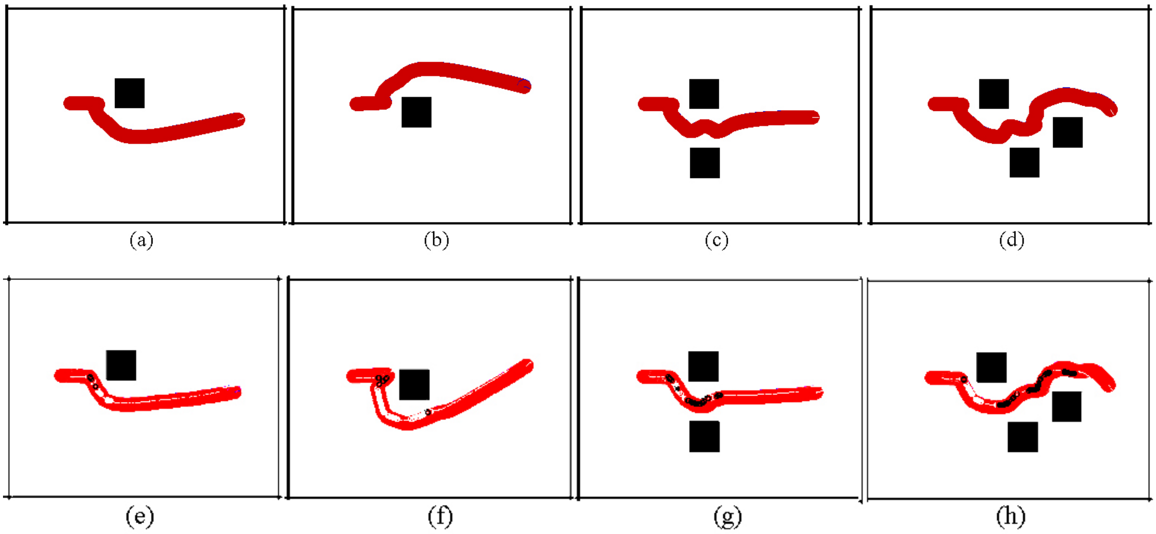

Fig. 4.a-d) robot trajectories returned by potential fields; e-f) robot trajectories returned by CBR trained with trajectory in Fig.4.a.

Since potential fields tend to return smooth trajectories unless at specific situations, a new set of experiments consisted of training the CBR only with potential fields. After the robot learnt cases during the trajectory in Fig. 4.a., we repeated all the experiments in Fig. 3 (Figs. 4.e-f). Again, some cases were mutated during these tests, but we purposefully avoided updating the casebase to test how the robot performs when training is not heavy. It can be observed that this time the robot does not get so close to obstacles and, hence, trajectories are smoother. Fig. 4.f shows an interesting case. The robot should leave the obstacle on the right to achieve a shorter trajectory, but since its training did not cover this situation, it chooses to stick to its knowledge and leave in on the left. When a human trained the robot, though, mutations made the robot work in other way (Fig. 3.b). This shows that the robot does not always learn the same.

Thus far, no knowledge has been injected to the robot farther than a trajectory leaving an obstacle on the left, either given by a human or by potential fields. Thus, our next experiment aims at testing how much further training improves the global performance of the agent. Fig. 5.a shows a trajectory after the robot has learnt all cases during the human guided training in Figs. 2.a-d. It can be observed that after all that training only a few new cases, marked in white, are acquired during this experiment, when the robot moves close to the first obstacle, mostly because human drivers tend to move close to obstacles, forcing the robot under its safety threshold and provoking mutations of the unwanted cases. It can be observed in Fig. 5.b, where the robot learnt its cases from a potential field approach, that almost no new cases are acquired when the safety threshold is kept. It can also be observed that in this case the trajectory presents less oscillations not only that the one returned by potential fields but also less than the one provided by CBR after training a single potential fields trajectory. Fig. 5.c shows the trajectory returned by the robot when it learns both potential field and human trained trajectories. In this case, black circles correspond to retrieved cases learnt from potential fields and white ones to those learnt from human drivers. It can be observed that the robot mostly prefers potential fields cases except when it needs to get really close to obstacles. It can also be observed that in this case the initial curvature change is smoother than in the previous ones.

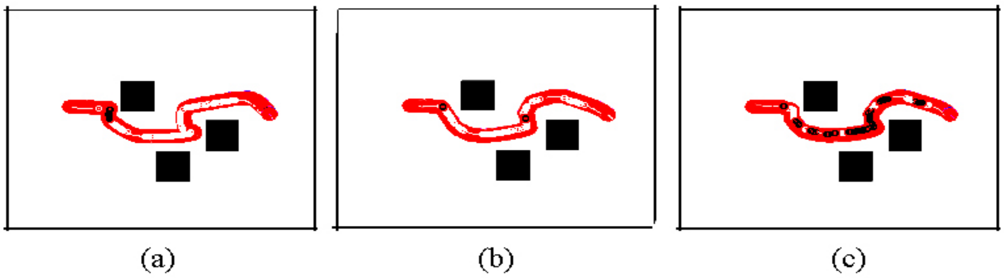

Fig. 5. Trajectory in a 3-obstacle environment when the robot has learnt all cases in: a) tests in Fig.2; b) tests in Fig. 4.a-d; c) tests in Figs.2 and 4.a-c.

Thus far, it has been tested that the trajectories returned by our CBR reactive scheme tend to be shorter and smoother than those returned by potential fields, but it is still necessary to test if the scheme can deal with cases which can not be solved by conventional purely reactive strategies. To this purpose, Fig. 6 present a new obstacle layout that has been constructed to prevent potential fields from finding a way through the corridor by moving obstacles progressively closer. Fig. 6.a shows how the potential field approach makes the robot stops at the beginning of the corridor. Similarly, Fig. 6.b shows the proposed scheme response when no cases are available at all. In this case, the robot reaches no equilibrium and instead keeps moving around looking for an entry in the corridor. Figs. 6.c-d show the trajectories returned by the proposed scheme after the human training if Fig. 2 and potential field based training in Figs. 4.a-d. In both cases, the robot moves through the corridor, even though no human training has prepared it for this specific test and potential fields could provide no trajectory to learn. Furthermore, if no training is available at all but we allow acquisition of new cases during the experiment in Fig. 6.b, the robot manages to find a way in on its own after a few turns (Fig. 6.e). It can be noted that the smoothest trajectory is returned after learning from potential fields but it is interesting to note that the trajectory returned when the robot learns on its own is also smoother than the one returned by the human trained robot. This occurs because the robot learns to move very close to obstacles from its human trainer and, hence, when the obstacle layout differs from the one in the training stage, in some cases it needs to change its trajectory in a sharp way to adapt itself to the current situation.

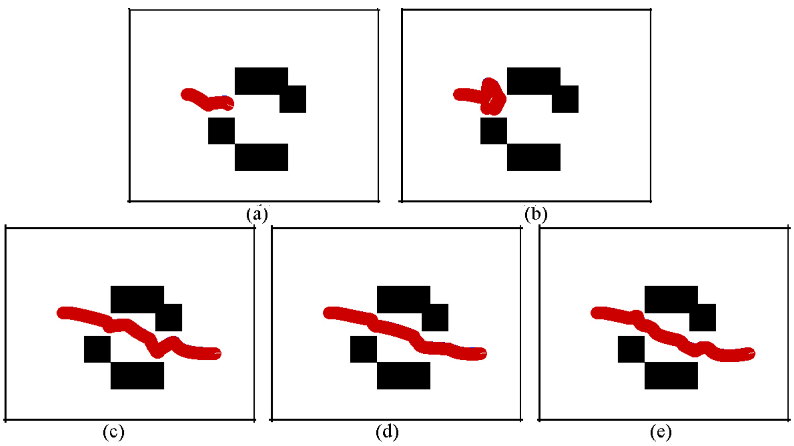

Fig. 6. Trajectories returned by: a) potential fields; and CBR with b) no training; c) human training; d) potential fields training; e) auto-training.

4. CONCLUSIONS AND FUTURE WORK

This paper has evaluated the learning capacities of a new CBR based pure reactive scheme for autonomous navigation. To keep a controlled environment for objective evaluation of the results, all tests have been performed under simulation. The system can acquire knowledge from human drivers, other reactive or non reactive navigation approaches and also from its own experience. Trajectories returned by the proposed system tend to be smooth, short and safe depending on the knowledge acquired by the robot: if the robot is taught to get close to obstacles, it does if necessary. It has been observed that the system acquires progressively less and less cases even if moving in different layouts. Also, a few cases are usually enough to provide efficient trajectories to a goal. Learnt cases tend to be adequate even when training involves bad cases as well. The scheme is capable of adapting to unexpected situations and also of improving the performance of the systems it used to learn new cases. Future work will focus on evaluating the performance of the scheme in real environments to test the effect of sonar errors.

5. ACKNOWLEDGEMENTS

This work has been partially supported by the Spanish Ministerio de Ciencia y Tecnologia (MCYT), project No. TIC2001-1758.

6. REFERENCES

[1] A. Aamodt, “Explanation-driven case-based reasoning”, in S. Wess, K. Althoff, M. Richter (eds.), *Topics in Case-based reasoning,* Springer Verlag, 1994, pp 274-288.

[2] S. Fox, and D.B. Leake, "Combining case-based planning and introspective reasoning", *Proc. of the 6th Midwest Artificial Intelligence and Cognitive Science Society Conference*, Carbondale, IL, 1995

[3] D. Fox, W. Burgard and S. Thrun, "The Dynamic Window Approach to Collision Avoidance", *IEEE Robotics & Automation Magazine*, Vol. 4, No. 1, 1997

[4] K.Z. Haigh and M. Veloso, "Route planning by analogy", *Proc. of Int. Conf. on Case-Based Reasoning*, Springer Verlag: Berlin, pp. 160-180, 1995

[5] Y. Koren and J. Borenstein, “Potential field methods and their inherent limitations for mobile robot navigation”, Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA), pp. 1398-1404, 1991

[6] J.L. Kolodner, “Case-Based Reasoning”, Morgan Kaufmann Publishers Inc., 1993.

[7] M. Kruusmaa, "Global navigation in dynamic environments using Case-Based Reasoning", *Autonomous Robots*, 14, pp. 71-91, 2003

[8] M. Likhachev, and R.C. Arkini, "Spatio-temporal case-based reasoning for behavioral selection", Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA), Seoul, Korea, pp. 1627-1634, 2001

[9] S. Quinlan and O. Khatib, "Elastic Bands: Connecting Path Planning and Control", *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 802-807, USA, 1993

[10] A. Ram and J.C. Santamaria, "A multistrategy case-based and reinforcement learning approach to self-improving reactive control systems for autonomous robotic navigation", Proc. of the 2nd Int. Workshop on Multistrategy Learning, Harpers Ferry, West Virginia, 1993

[11] I. Ulrich and J. Borenstein, "VFH*: Local Obstacle Avoidance with Look-Ahead Verification", *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2505-2511, San Francisco, CA, April 2000

[12] C. Urdiales, E.J. Pérez, F. Sandoval, J. Vázquez-Salceda, *“A CBR based pure reactive layer for autonomous robot navigation”*, 7th IASTED Int. Conf. on Artificial Intelligence and Soft Computing (ASC 2003), Banff (Canada), July 2003.