

A CBR approach to Behaviour-Based Navigation for an Autonomous Mobile Robot*

Alberto Poncela, Cristina Urdiales and Francisco Sandoval

*Departamento de Tecnología Electrónica
E.T.S.I. Telecomunicación
Universidad de Málaga,
Campus de Teatinos, 29071 Málaga, Spain
apg@dte.uma.es*

Abstract – This paper presents a new CBR navigation scheme based on three behaviours: Wall Following, Corridor Following and Door Crossing. The system switches among CBR behaviours depending on a local grid built around the robot from sonar sensor readings. The main advantage of the proposed system is that it can easily adapt to any robot and environment, learning, both by training and by its own experience. The method has been successfully tested in real environments with a Pioneer robot equipped with 8 frontal sonar sensors.

Index Terms – Behaviour, navigation, learning, case-based reasoning, robot.

I. INTRODUCTION

Autonomous navigation is one of the main issues a mobile robot aims at. It is defined as the problem of reaching a defined goal in a safe way. Navigation may be achieved through a behaviour-based control, that has been established to serve as building blocks for robotic actions [2].

According to Mataric [10], a robot behaviour is a control law that satisfies a set of constraints to achieve and maintain a particular goal. There are two main approaches to deal with robot behaviours in a behaviour-based navigation scheme: behaviour coordination and behaviour competition. Behaviour coordination mechanism ensures simultaneous activity of several independent behaviours and obtains a coherent behaviour that yields to the intended navigational task [14]. Its main difficulty is to find the cooperation rule to fuse behaviours in an efficient way, because it tends to cause the robot to be indecisive when the behaviours have mutually exclusive interest with nearly equal importance [19]. To that purpose, different techniques have been employed, such as the fusion of control signals addressed in [4]. Fuzzy logic has also been reported to do well in real unknown environments [14] [12]. Other methods prefer to combine fuzzy logic with neural network systems [7], or try to emulate the stimulus-response behaviour of a biological system [16].

On the other hand, when using competitive behaviours the system has to elicit one from the set of them to perform the navigation task. The issue is to choose the correct behaviour that better fits the environment at each moment. Thus, a switching method among behaviours has to be

employed. To that purpose, Zalama [22] proposes a model that is made up of several interconnected modules which allow the robot to obtain the linear and angular velocities. Rusu [17] employs command fusion to combine the output of several neuro-fuzzy subsystems for sensor-based navigation in indoor environments, but only simulated results are presented.

When dealing with dynamic changing environments, behaviour-based systems need to adapt. However, changes are difficult to model and predict. The main drawback of modeling is the use of parameters to characterize kinematics and dynamics. These parameters need to be optimized for each specific problem, specially if different robots are used. Furthermore, if the robot is affected by physical problems, the same parameter optimization has to be used. Hence, it would be desirable to achieve a behaviour-based scheme able to adapt to changing circumstances without human supervision, allowing the system to work in a different robot after minor changes.

In this context, Case-Based Reasoning (CBR) emerges as an alternative for adapting to environment changes. CBR is a learning and adaptation technique to solve current problems by retrieving and adapting past experiences [1]. As demonstrated, when using CBR there is no need to study the robot kinematics nor the environment [20].

In this paper we propose a CBR navigation scheme based on three behaviours: Wall-Following, Corridor-Following and Door Crossing. This navigation strategy is supported by a behaviour selector which switches among behaviours depending on a local grid built from sonar readings [13]. The paper is organized as follows: Section II presents the structure of the proposed behaviour-based system. Section III describes the CBR implementation of behaviours. Section IV shows the experimental results. And finally, conclusions are presented in Section V.

II. BEHAVIOUR-BASED NAVIGATION SYSTEM

The proposed behaviour-based system is made up of several interconnected modules which allows the mobile robot to obtain the translational and rotational velocities that must be applied for navigation at any time. The modules of the system are presented in Fig. 1:

* This work has been partially supported by the Spanish Ministerio de Educación y Ciencia (MEC) and FEDER funds, project No. TIN2004-07741 and project No. TIN2005-01359.

1. *Sensors*. It provides sonar readings for two purposes: i) build the local grid; ii) supply reactive information to behaviours to maintain an actualized knowledge of the closest environment.
2. *Local Map*. It is a local grid built around the robot from sonar readings (Sensors). This grid is fast and easy to construct, efficiently combining any number of sensor readings.
3. *Behaviour Selector*. Its main function is to switch among behaviours. The switching criteria is based upon a place recognition technique presented in [13].
4. *Behaviours*. It is composed of three submodules, one for each of the three competitive behaviours that have been implemented in the system. It receives the switching information from the Behaviour Selector module to activate one of the behaviours. Its output is the translational and rotational velocities that must be applied to the robot.
 - a) *Wall Following*. This first behaviour tries to move the robot around the environment following the contour of its right wall at a predefined distance.
 - b) *Corridor Following*. This behaviour allows the robot to go across a corridor by its centre.
 - c) *Door Crossing*. The last behaviour permits the robot to cross a door to come in or come out of a room.
5. *Motors*. This module receives the translational and rotational velocities that actually are applied to the robot motors.

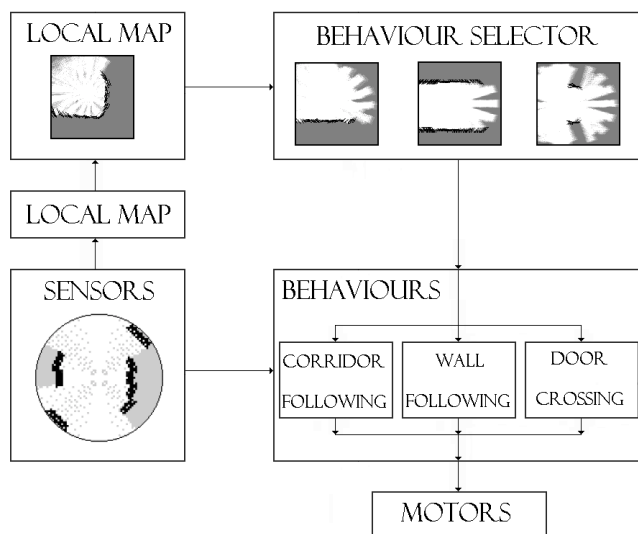


Fig. 1 Behaviour-Based Navigation System.

Behaviours module receives switching information from the Behaviour Selector. By default, Corridor Following Behaviour is activated. When the Behaviour Selector module detects a wall in front of the robot in the Local Map, it forces to switch to Wall Following. Otherwise, if the Local Map presents a door pattern, the Door Crossing behaviour is selected. Due to its modularity, the system can be extended to more complex behaviours, by detecting new patterns in the local grid. However, really complex behaviours should be achieved via a deliberative layer supported through a hierarchical architecture to avoid loss of generality [21].

III. CBR APPROACH TO BEHAVIOUR-BASED NAVIGATION

CBR is a learning and adaptation technique to solve current problems by retrieving and adapting past experiences. CBR stores any possibly interesting situation in a casebase in the form of cases. A CBR case is a N-dimensional input vector to characterize a given situation and the solution to that situation. The advantage of CBR compared to another techniques, such as neural networks, is that cases in the casebase are explicitly stored. Thus, cases can be easily analyzed to have a clear idea of what the robot has learnt and why it performs a given action. Furthermore, learning through CBR is preferable than neural networks since it is possible to seed the casebase with a-priori knowledge [8], and the training stage can be done from a small number of samples.

A CBR cycle to solve a new situation consists of: i) retrieve the most similar case stored in the casebase to the current case; ii) adapt its stored solution to the current case; iii) evaluate the efficacy of the case to become the solution of the current case; iv) learn from the new experience. Therefore, when designing a CBR system it must be considered: i) the description of the problem; ii) the case instance; iii) how to retrieve a stored case evaluating the similarity between cases; iv) how to adapt the retrieved solution to solve a new situation; and v) how to learn new experiences.

CBR has previously been used in robot navigation. Most approaches deal with static environments [3]. There are also some approaches to deal with dynamic environments, using a topological map of an a-priori known environment [5], although the topological map has to be reorganized when new opportunities are discovered. To overcome the problem stated, Kruusmaa proposes to use grids [8], but she concludes that CBR-based global navigation in beneficial only when obstacles are large and dense. There are also some approaches that concerns about CBR behaviour-based navigation (e.g. [15]). Likhachev [9] proposes and scheme that relies on accumulating experience over a time window while navigating in a given environment to make behaviours cooperate. However, this scheme is not adequate in dynamic environments where the layout changes frequently.

In this work, we propose a CBR navigation scheme based on the three behaviours shown in Fig. 1. The casebase acts as as collection of sense/act pairs, in a reactive sense. The approach is valid for any environment, because it does not store information about any given environment, but reacts at a given time instant where there are obstacles at certain positions. Furthermore, the CBR system: i) adapts dynamically to changes; ii) is able to learn peculiarities of the environment; and iii) eliminates the need to study the robot kinematics.

A. Case Instance

Fig. 2 presents the parameters to define our case. The input of the proposed case instance is defined by the instant sensor readings. The output of the case consists of the

translational velocity (v_r) and the rotational velocity (v_t) to apply to the robot. These instant velocities implement the desired behaviour. Thus, they may vary depending on the selected one. The heading of the robot has not been included in the case instance because it does not affect to the problem since the robot must react to an obstacle configuration independently of its orientation.

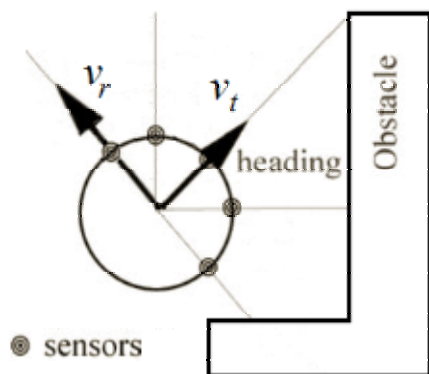


Fig. 2 Case definition.

B. Retrieval process

Minor differences among sensors readings may lead to different cases. However, these differences usually correspond to same situations. Since it has been proven that it is better to combine discrete and continuous attributes in CBR systems [18], the problem instance can be improved by discretizing the sensor reading. Hence, the method discretizes 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); v) no influence (more than 150 cm). This discretizing process assumes for our reactive behaviours that far obstacles do not influence while the closer an obstacle is, the more dangerous the situation is.

Our casebase is organized as a flat memory. Thus, the retrieval process consists of matching all cases in the casebase against the current one. The most similar case is selected evaluating the similarity between cases through a discretized Manhattan distance. Let N be the number of input sonar readings of the problem case instance. Let $\vec{X} = [x_1, x_2, \dots, x_N]$ and $\vec{Y} = [y_1, y_2, \dots, y_N]$ be two vector of sonar readings. Assuming that the vector with the number of intervals for each sonar is $\vec{R} = [r_1, r_2, \dots, r_N] = [5, 5, \dots, 5]$, the used discretized Manhattan distance is defined as:

$$D(\vec{X}, \vec{Y}) = \frac{\sum_{k=1}^N w_k \times \left\lceil \frac{|x_k - y_k|}{r_k - 1} \right\rceil}{\sum_{k=1}^N w_k} \quad (1)$$

where $\vec{W} = [w_1, w_2, \dots, w_N]$, is the vector of weights for sonars. The larger the weight of a sonar, the more is its influence in the retrieval process. We have considered that

every weight is equal to one, so that we do not favour any direction rather than others.

C. Behaviour Training, Learning and Adaptation

In any CBR system, learning is one of the most important tasks. There are two main learning approaches: learning by observation and learning by own experience. Learning by observation happens when the case library is seeded with a set of initial cases. Learning by own experience is done after each cycle of the reasoner. In our behaviour-based system, both approaches are used to learn each of our three implemented behaviours (Wall Following, Corridor Following and Door Crossing), separately.

An initial off-line stage is used to learn by observing a human, who guides the robot using a keyboard to achieve the desired behaviour. This supervised learning has the advantage of adapting to any situation. Also, the human driver considers the kinematics and dynamics of the problem. During this stage, the robot captures its instant sensory readings to build a case each time a different sensory pattern is detected. These cases are stored in the case library, together with their solution.

To reduce the time of the retrieval process, the casebase obtained through this off-line stage is organized by dividing it into classes using a MaxMin algorithm. Each class has a prototype. These prototypes become the cases of our casebase. Let S be the number of stored cases in the casebase. Let C be the number of classes which the casebase has been splitted. Then, the prototype of a class i is:

$$P_i = \sum_{\substack{k=1 \\ \forall k \in i}}^S Case_k \quad (2)$$

It is necessary to note that this clustering stage averages cases of every class. Furthermore, the algorithm only depends on a threshold T_{max} , which controls the maximum size of our classes. If T_{max} is high, the number of classes is small and the casebase presents a few cases. If it is low, the casebase has more cases (classes). However, its selection is not critical and can be broadly chosen. In this paper, our experiments were carried out with $T_{max} = 1000$. This leads to less than 100 classes.

While the robot is moving around the environment performing the selected behaviour, it learns by its own experience. When the sensory pattern changes, the system retrieves the most similar case in the casebase. Then, if the discretized Manhattan distance between the current case and the retrieved one is lower than a threshold T_{adapt} , the solution of the retrieved case is adopted. Otherwise, a new situation has been detected. Therefore, the case is adapted, used and stored.

To solve the new detected situation, the system relies on a derivational adaptation [1]. It obtains a new solution for the current case using an analytical model. It would not be possible, for example, to take the two nearest cases and merge their solutions into a new case, since these two cases would not fit the environment layout. The model is based on

an addition of forces that implements behaviours. For each behaviour, the rotational velocity is calculated as:

1. *Wall Following*. This behaviour is implemented adding three forces. The first (f_{paral}), maintains the robot parallel to the wall. The second (f_{dist}), tries to maintain the robot at a predefined distance of the right wall. And the third (f_{avoid}), avoids obstacles that appear while the robot is navigating. The expression of the rotational velocity is:

$$v_r = f_{\text{paral}} + f_{\text{dist}} + f_{\text{avoid}} \quad (3)$$

2. *Corridor Following*. It is based on the same philosophy of the former behaviour, adding three forces. As in the Wall Following behaviour, f_{paral} tries to move the robot parallel to a wall. However, f_{dist} does not try to maintain the robot at a distance of the wall, since it should move by the centre of the corridor. Thus, f_{dist} takes into account its wideness. Finally, f_{avoid} avoids obstacles that may appear in the way of the robot.

$$v_r = f_{\text{paral}} + f_{\text{dist}} + f_{\text{avoid}} \quad (4)$$

3. *Door Crossing*. In this case, only two forces are added. The first one is f_{perpen} . This force allows the robot to stay perpendicular to the mark of the door. The last one (f_{avoid}), avoids the mark of the door, preventing the robot to collide with it.

$$v_r = f_{\text{perpen}} + f_{\text{avoid}} \quad (5)$$

The translational velocity is obtained in the same manner for our three behaviours. Let $v_{t \text{ max}}$ and $v_{r \text{ max}}$ be the maximum translational and rotational velocities, respectively. Then, the translational velocity is:

$$v_t = v_{t \text{ max}} \left[1 - \frac{v_r}{v_{r \text{ max}}} \right] \quad (6)$$

The higher the rotational velocity, the lower the translational one. When the rotational velocity is maximum, the translational velocity is null. Hence, when the robot is rotating at a high speed, it tends to translate at a low velocity. In this paper, our experiments were realized with $v_{r \text{ max}}=15$ %/s and $v_{t \text{ max}}=0,1$ m/s.

IV. EXPERIMENTS AND RESULTS

The proposed system has been tested on a Pioneer robot equipped with 8 frontal Polaroid sonar sensors. The robot builds an evidence grid [11] to show where obstacles are. In these grids, free space is printed in black, obstacles in white and unexplored areas are presented in gray. It is necessary to point out that grids are simply used to graphically represent the environment.

Before testing the system in the real robot, we run several tests under simulation to keep the environment under control. Their aim is to evaluate the learning and adaptation process. T_{adapt} was fixed to 0,1.

Fig. 3 and Fig. 4 show the training and adaptation process of our Wall Following behaviour. The environment where the robot was trained is presented in Fig. 3a. The goal of the robot is to follow the wall on its right. It is initially heading right. After human guiding the robot (Fig. 3b), it has learnt to follow the wall turning left by observation, storing

this knowledge in a casebase. The learning process continues in the environment of Fig. 4a. The robot is in the top area of the middle obstacle, heading right. It starts to follow the wall as presented in the grid of Fig. 4b. As the situation has been previously trained and stored in the casebase, the robot applies the solution of the most similar stored case to perform the behaviour. When it reaches the position in Fig. 4c, the robot detects this is a new situation since the distance to the most similar case in the casebase is higher than T_{adapt} . As we can observe in Fig. 4c, this happens because the robot has been trained to turn left, and in the new situation the robot must turn right to follow the wall. Thus, the case is adapted, used and stored, learning by its own experience. When the robot is in the position of Fig. 4d, the robot finds a familiar situation, following the wall on the right. It must be noted that the robot has not been trained to perform a right turn at the end of the trace in Fig. 4a. However, the CBR system can resolve this case because it has been previously learnt through the adaptation process.

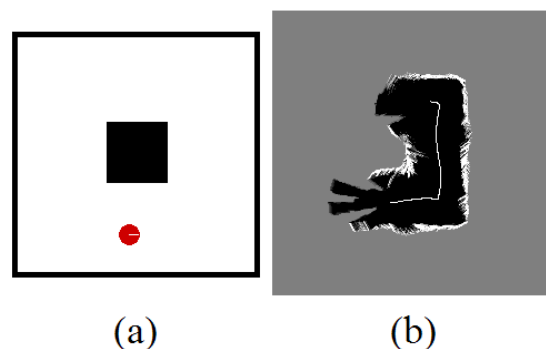


Fig. 3. Wall Following Training: a) scenario; b) trace.

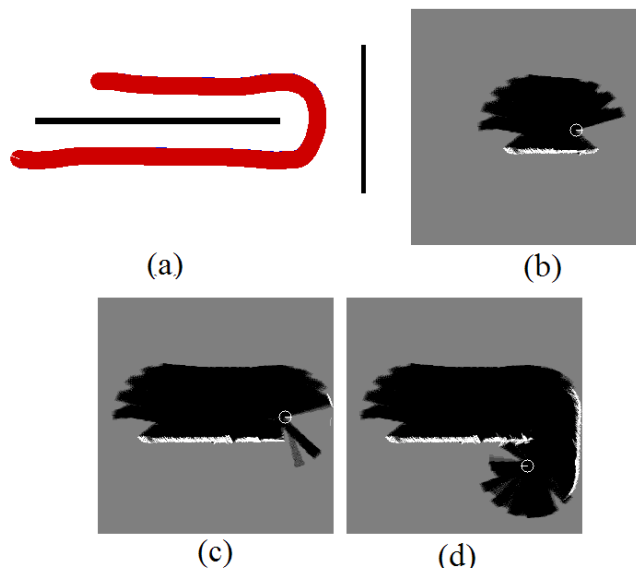


Fig. 4. Wall Following Adaptation: a) trace; b) trained situation; c) new situation, adaptation; d) trained situation.

First, the real robot was supervisedly trained so that it learnt by observation. Fig. 5 shows the trace of the robot when a human guides it to learn the three behaviours. It must be noted that this off-line stage is quite brief. However, this

is not a drawback since the CBR system can learn by its own experience while adapting to new situations.

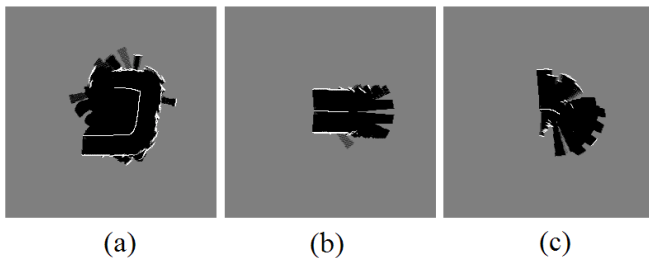


Fig. 5. Supervised training : a) Wall Following; b) Corridor Following; c) Door Crossing.

Fig. 6 shows the performance of the robot in a real test environment. The scenario consists of a corridor and a door, as we can observe in the sight of the environment in Fig. 6a. The door is approximately 90 cm wide, conforming a 45° angle with the corridor. The robot is initially located at point A (Fig. 6a), heading north. Since the system detects the robot is in a corridor (Fig. 6b), the Corridor Following behaviour is activated at the beginning. The robot continues following the corridor until it reaches the position marked as point B. At this instant, the system detects a door pattern in the local grid of Fig. 6c. Therefore, the Door Crossing behaviour is activated, allowing the robot to cross the door it has found.

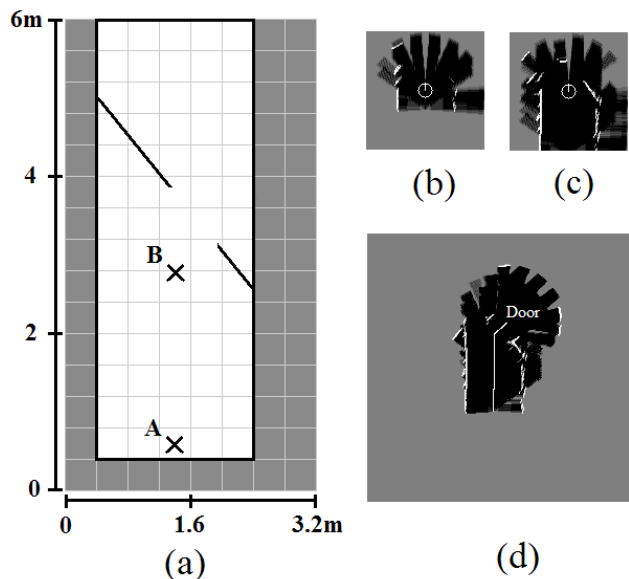


Fig. 6. Real environment 1: a) sight of the environment; b) Corridor Following behaviour pattern; c) Door Crossing behaviour pattern; d) trace.

The trace of the robot is superimposed in white color to the evidence grid built during the movement (Fig. 6d). It must be noted in this trace that: i) the robot follows the corridor from point A to point B in almost a straight way; ii) the door is crossed by its middle without colliding with its marks. Although the system is supported by the stored training, it also learns by experience through the adaptation process while performing both behaviours. Thus, as long as this experience is stored in the casebase, the system progressively adapts to each of the behaviours.

A second test environment is presented in Fig. 7. It is made up of two corridors and a door. The first corridor is 1.3m wide. The second is a little wider, 1.8m. The door is situated perpendicular to the second corridor, presenting a 90 cm wideness. However, it is not centred on it. This door is closer to the left wall (40cm) than to the right wall (50cm).

The robot starts at point marked as A in Fig. 7, heading north. The Behaviour Selector module detects a Corridor Following pattern at this position (Fig. 8a). Hence, the robot starts moving by the centre of the first corridor. When the autonomous agent arrives at point B, the Behaviour Selector module switches to Wall Following, since the local grid presents this pattern (Fig. 8b). However, this behaviour is deactivated as soon as the system reaches point C in Fig. 7. At this position, the system detects the pattern of the second corridor (Fig. 8c). Finally, the robot must cross the door. This is achieved by selecting the Door Crossing behaviour when it is at point D.

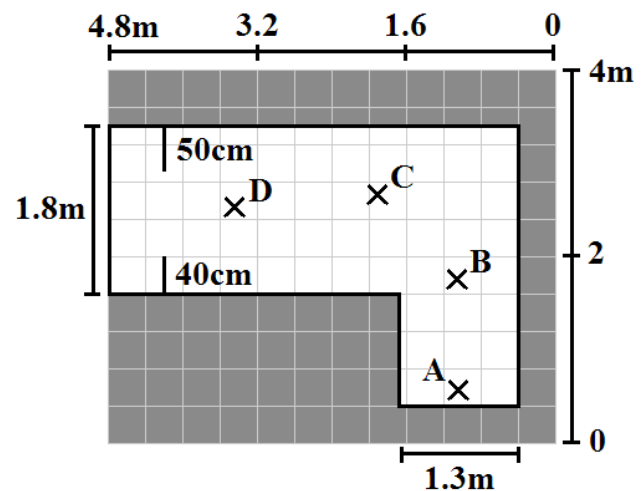


Fig. 7. Sight of real environment 2.

The trace followed by the robot in the environment of Fig. 7 is presented in Fig. 8e. Again, it is superimposed in white color to the built evidence grid. The robot follows both corridors in almost a straight manner. While the robot follows the first corridor, the north wall of the environment is detected. Thus, as stated before, the system switches to follow this wall. This behaviour tries to maintain the robot parallel to the right wall, until the second corridor appears. After that, the robot centres in this corridor. Finally, the system switches to Door Crossing. The robot crosses the door by its centre. However, the door is not symmetrically centred in the corridor. Therefore, the agent approaches to the left wall, as we can see in Fig. 8e. Obviously, while moving, the robot uses trained situations, as well as it adapts to new ones. During the experiments, the robot learnt about 40 new cases related to new situations for each of the behaviours.

It is necessary to point out that there have not been formulated any explicit consideration about hardware structure, kinematics, sensor errors or environmental factors. Hence, if any of these parameters changes, the system adapts to the new situation. This allows the system to be valid for

different robot platforms. Furthermore, the system has been compared to a reactive scheme, the Potential Field Method [6]. Our system: i) presents less oscillations; ii) avoids the difficulty to move between close obstacles (e.g., doors).

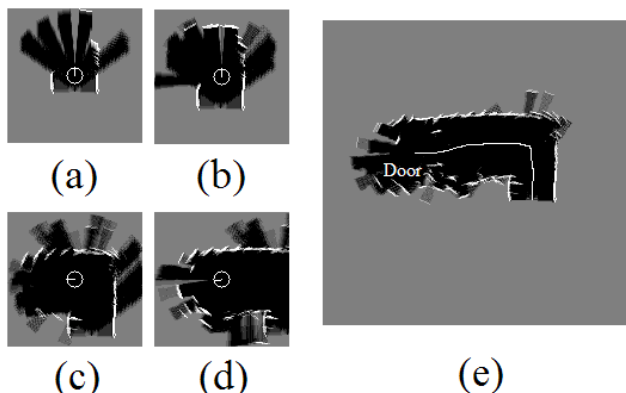


Fig. 8. Real environment 2: a) Corridor Following behaviour pattern; b) Wall Following behaviour pattern; c) Corridor Following behaviour pattern; d) Door Crossing behaviour pattern; e) trace.

V. CONCLUSIONS

This paper has presented a new CBR scheme to behaviour-based navigation for a mobile robot. The proposed system is based on three behaviours: Wall Following, Corridor Following and Door Crossing. These behaviours are competitive. The system switches among behaviours depending on a local grid built from sonar sensor readings. The system learns by observation of a human driver and learns by its own experience. Furthermore, it is able to adapt to changes with no need to study the environment layout or the robot kinematics. The proposed behaviour-based system has been thoroughly tested on a Pioneer robot equipped with 8 frontal sonar sensors. Future work will focus on increasing the number of behaviours that can be selected, and on fusing behaviours to make them cooperate.

REFERENCES

- [1] A. Aamodt and E. Plaza, "Case-Based Reasoning: Foundational Issues Methodological Variations, and System Approaches", *AI Communications*, vol. 7, no. 1, pp. 39-59, March, 1994.
- [2] R.C. Arkin, *Behaviour-Based Robotics*, MIT Press, Cambridge, Massachusetts, 1998.
- [3] S. Fox and D.B. Leake, "Combining case-based planning and introspective reasoning", *Proceedings of the 6th Midwest Artificial Intelligence and Cognitive Science Society Conference*, Carbondale, United States, 1995.
- [4] E. Freire, T. Bastos-Filho, M. Sarcinelli-Filho and R. Carelli, "A New Mobile Robot Control Approach via Fusion of Control Signals", *IEEE Transactions on Systems, Man and Cybernetics-Part B: Cybernetics*, vol. 34, no. 1, pp. 419-429, 2004.
- [5] K.Z. Haigh and M. Veloso, "Route Planning by Analogy", *Proceedings of the International Conference on Case-Based Reasoning*, Springer Verlag, Berlin, pp. 160-180, 1995.
- [6] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots", *International Journal of Robotics Research*, vol. 5, no. 1, pp. 90-98, 1989.
- [7] S. Khatoun and S. Khatoun, "Behavior Coordination of Autonomous Mobile Robot Navigation by Neuro-Fuzzy Systems", *Proceedings of the IEEE 31st Annual Northeast Bioengineering Conference*, pp. 56-60, United States, 2005.
- [8] M. Kruusmaa, "Global Navigation in Dynamic Environments using Case-Based Reasoning", *Autonomous Robots*, vol. 14, pp. 71-91, 2003.
- [9] M. Likhachev and R.C. Arkin, "Spatio-Temporal Case-Based Reasoning for Behavioral Selection", *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA01)*, pp. 1627-1634, Seoul, Korea, 2001.
- [10] M.J. Mataric, "Behavior-based Control: Examples from Navigation, Learning and Group Behavior", *J. Experimental Theoretical Artificial Intelligence, Special Issue: Software Architecture Physics Agents*, vol. 9, pp. 46-54, 1997.
- [11] H.P. Moravec, "Sensor Fusion in Certainty Grids for Mobile Robots", *AI Magazine*, vol. 9, pp. 61-74, 1998.
- [12] S. Parasuraman, V. Ganapathy and B. Shirinzadeh, "Behaviour Based Mobile Robot Navigation Technique for Real World Environments using Fuzzy Logic System", *2004 IEEE International Conference on Systems, Man and Cybernetics*, pp. 3359-3364, The Netherlands, 2004.
- [13] A. Poncela, C. Urdiales, C. Trazegnie and F.Sandoval, "A New Sonar Landmark for Place Recognition", *Proceedings of the 6th International FLINS Conference (FLINS2004)*, Blankenberge, Belgium, 2004.
- [14] N. Rahman and A.R. Jafri, "Two Layered Behaviour Based Navigation of a Mobile Robot in an Unstructured Environment using Fuzzy Logic", *IEEE 2005 International Conference on Emerging Technologies*, pp. 230-235, Islamabad, Pakistan, 2005.
- [15] A. Ram, R.C. Arkin, K. Moorman and R.J. Clark, "Case-Based Reactive Navigation: A Method for On-Line Selection and Adaptation of Reactive Robotic Control Parameters", *IEEE Transactions on Systems, Man and Cybernetics-Part B: Cybernetics*, vol. 27, no. 3, pp. 376-394, 1997.
- [16] B.B.K. Reddy, B. Kimiaghdam and A. Homaifar, "Reactive Real Time Behavior for Mobile Robots in Unknown Environments", *2004 IEEE International Symposium on Industrial Electronics*, pp. 693-697, Ajaccio, France, 2004.
- [17] P. Rusu, E.M. Petriu, T.E. Whalen, A. Cornell and H.J.W. Spoelder, "Behavior-Based Neuro-Fuzzy Controller for Mobile Robot Navigation", *IEEE Transactions on Instrumentation and Measurement*, vol. 32, no. 4, pp. 1335-1340, 2003.
- [18] M. Sánchez-Marré, U. Cortés, J. Béjar, I.R. Roda and M. Poch, "Reflective Reasoning in a Case-Based Reasoning Agent", *Lecture Notes in Artificial Intelligence*, vol. 1624, pp. 143-158, Springer-Verlag, 1999.
- [19] M.F. Selekwa, D.D. Dunlap and E.G. Collins, "Implementation of Multi-valued Fuzzy Behavior Control for a Robot Navigation in Cluttered Environments", *Proceedings of the 2005 IEEE International Conference on Robotics and Automation (ICRA05)*, pp. 3688-3695, Barcelona, Spain, 2005.
- [20] C. Urdiales, E.J. Pérez, J. Vázquez-Salceda, M. Sánchez-Marré and F. Sandoval, "A Purely Reactive Navigation Scheme for Dynamic Environments using Case-Based Reasoning", *Autonomous Robots*, vol. 21, pp. 65-78, 2006.
- [21] C. Urdiales, A. Bandera, E.J. Pérez, A. Poncela and F. Sandoval, "Hierarchical planning in a mobile robot for map learning and navigation", in *Autonomous Robotic Systems – Soft Computing and Hard Computing Methodologies and Applications*, D. Maravall, D. Ruan and C. Zhou, Eds., Physica Verlag, pp. 165-188, 2003.
- [22] E. Zalama, J. Gómez, M. Paul and J.R. Perán, "Adaptive Behaviour Navigation of a Mobile Robot", *IEEE Transactions on Systems, Man and Cybernetics-Part A: Systems and Humans*, vol. 32, no. 1, pp. 160-169, 2002.