# AUTONOMOUS MONITORING AND REACTION TO FAILURES IN A TOPOLOGICAL NAVIGATION SYSTEM

V. Egido, R. Barber, M. J. L. Boada, M. A. Salichs
*Robotics Lab. Carlos III University.Madrid, Spain*

Keywords: Robot navigation; Monitoring; Supervision; Robot skills.

Abstract: In this paper a system for simultaneous navigation and monitoring with autonomous reaction to failures is going to be presented. This system is part of a complete navigation system called AURON (Autonomous Robot Navigation). The AURON System autonomy is based on the interaction of four main components: the autonomous generation of an environment representation, the planning of a sequence of actions and perceptions which guide the robot from an initial event to a final one, the navigation that converts sequences in real movements and supervises all the process, and the relocalization that allows to place the robot again in the representation. This system has been implemented in a mobile robot control architecture called AD. AD is a two level architecture: deliberative and automatic. The paper is focused in one deliberative skill, the navigation skill.

## 1 INTRODUCTION

Autonomous movement implies not only being able to perform a priori established movements without human help, but also to react to unexpected situations, with the same ability. Reacting to unexpected events is a skill related to intelligence.

To supervise, will then require to monitor the system evolution and to choose the most suitable action over the events that halt the plan execution. As for the monitoring techniques, there is no generally accepted definition of execution monitoring (Fichtner, 2003). Giacomo et al in (Giacomo, 1998) defined it as ‘the robot’s process of observing the world for discrepancies between the actual world and the robot’s internal representation of it, and recovering from such discrepancies’.

The recovering techniques that Giacomo describes are widely dependant on the navigation system architecture being used. The purely reactive systems, like the one develop by Brooks (Brooks, 1986), do not have and cannot contain recovering techniques because its basis are the reaction to events without any kind of deliberative capacity over themselves.

On the other hand, within the architectures which consider those techniques because they contain a deliberative character, the works (Stuck, 1995), (Alami, 1998), (Fernandez, 1998) and (Fichtner, 2003) are found. All of them have a level over the purely reactive one, which allows supervising the navigation process, specifying which types of situations are found, trying to correct them, recovering the system control and, if it is possible, accomplishing the required task.

In this work, a navigation system which monitors what is happening and reacts to changes, is presented. This system is called AURON and has been implemented in the hybrid architecture AD (Barber, 2001).

## 2 THE AURON NAVIGATION SYSTEM

AURON (Autonomous Robot Navigation) is considered a complete navigation system formed by the interaction of four main components. As it is shown in figure 1, the components are the explorer, the planner, the navigator and the relocalizator.



Egido V., Barber R., J. L. Boada M. and A. Salichs M. (2005).
AUTONOMOUS MONITORING AND REACTION TO FAILURES IN A TOPOLOGICAL NAVIGATION SYSTEM.
In *Proceedings of the Second International Conference on Informatics in Control, Automation and Robotics - Robotics and Automation*, pages 383-386
DOI: 10.5220/0001156603830386

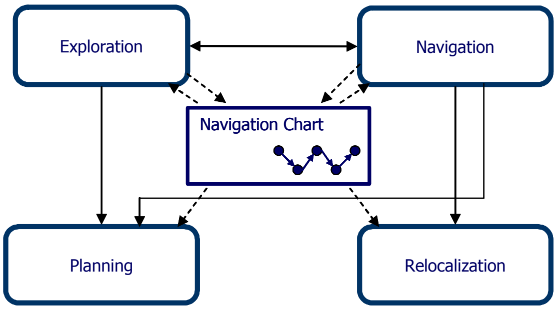


Figure 1: General structure of the AURON System

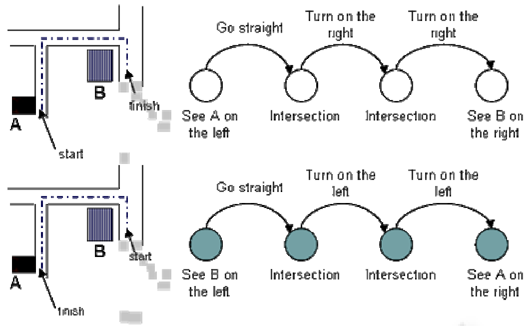


Figure 2: Example of Navigation Chart

Each component has its own mission and interacts with the others giving autonomy to the robot. The explorer is in charge of generating without human intervention the representation of the environment navigating through unknown places and obtaining information. The planner gives the sequence of actions and detected events to go from one place to another once the robot has its own representation. The navigator converts the sequence obtained by the planner in real movements and supervises all the process and finally, the relocalizator allows placing the robot again in the representation of the environment once it is lost.

The environment representation is an essential part of the system as it is used by all the other components. This representation is called the Navigation Chart. The Navigation Chart is a fundamentally topological representation of the environment, which tries to imitate the human navigation. It differs from other representations (Beccari, 1997) in considering a directed graph where arcs and nodes are equally relevant. In this graph arcs are not unions without information as in many other models and nodes have parameters that allow dynamic planning mechanisms different from previous developments.

The Navigation Chart is not formed by a succession of places of the environment as in (Remolina, 2004) but by a succession of elementary skills (Egido, 2003). It is represented by a simple directed parameterised graph G(**v**,**e**) formed by nodes **v** and edges **e**. Nodes are events regarding sensorial perceptions (be in front of a door) and edges represent sensorimotor skills which lead the robot to the next event (move towards a door). A new situation is being described on the Navigation Chart when detecting an event. The robot finds itself in the specific situation or place in which a specific sensorial event is sensed. An example of this Navigation Chart and the importance of its edges and nodes is shown in figure 2.

# 3 SUPERVISION

Once the sensorimotor and sensorial event skills to be detected are obtained, the skill carries out the supervision of what is taking place while those skills are activated or deactivated. To perform this supervision, the information related to nodes and edges is used. In this case, time and distance are parameters which allow supervising the process in a higher level, but also in some situations specific elements associated with enabled skills and events will be supervised.

Therefore, a two level supervision will be considered: A general level which implies the distance and time monitoring, which can be applied to all skills, and a lower level that represents a specific monitoring for specific events. This paper will focus in the general monitoring level, which can be applied to the skills used.

## 3.1 Distance monitoring

While the skills are enabled and disabled, the process will be supervised comparing the distances stored in the graph's edges with the real travelled distances. When there is a difference in percentage between the distance travelled by the robot and the one stored greater than a established value (for the environment where the experiment test were carried out, it has been empirically established that the distance travelled will not overcome in a 10% the one stored), the supervisor compares the actual sensorimotor skill and the one after the node detection having in mind two fundamental cases:

- The undetected event is not a decisive event for the plan execution.
- The undetected event sets a change in the execution plan and therefore it is relevant.

In figure 4 it is shown how the plan could continue if the event "Right door detected" failed to go from node A to node B, hiding the edge that implies a different action: "Cross door".

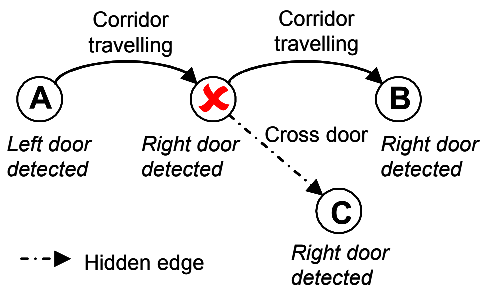


Figure 4: Planning with detection problems

In the second case, the sensorimotor skill that was being carried out and the next one are different. The event detection is totally necessary to continue with the plan. In figure 5 it would execute the plan that goes from node A to node C. In this case, the navigator hides the different outgoing edges of the failed node and communicates to the main sequencer the new situation carrying out a new plan.

This new plan, if possible, will give a new subgraph with the edges and nodes sequence that allows reaching the final node in the new situation.

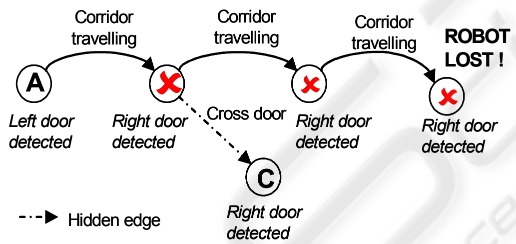


Figure 5: Planning with detection problems

To finish this supervision method monitoring distances, a last situation is considered. The robot could not detect the new node from the alternative plan. (Situation of 3 consecutive failures). In this case, it cannot be assured that the robot is in the place set by the Navigation Chart. If the robot is in this situation it will try a new plan, but if it fails, the navigator will indicate by an event that the robot is definitely lost and that there are not enough probes to consider that the robot is in the node set by the Navigation Chart. In figure 5 it can be seen how three consecutive event failures imply an event in which the navigation skill shows that the robot is lost.

## 3.2 Time monitoring

In all the cases described above, failures in events are monitored, but the environment changes could equally affect the sensorimotor skills. If the distance travelled is not overcame and however the time is overcame without detecting an expected event, then this means that a sensorimotor skill execution problem is found and it is halted in a Navigation Chart node.

The Navigation Chart characteristics that are being used, in most cases, make the Navigation Chart a graph that contains associated symmetries. An example of this is the fact that travelling the corridor in one way has its equivalent on travelling the corridor in the other way. A 180º robot turn allows travelling the corridor in the other way and finds an event that will allow localization, as it is shown in figure 6.

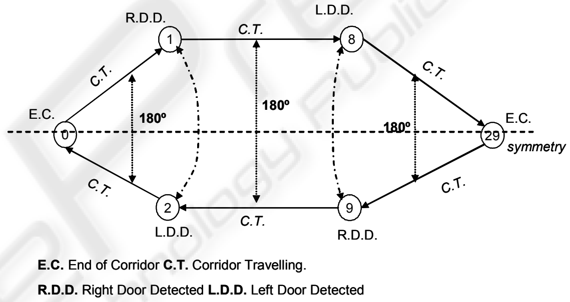


Figure 6: Graph symmetries to apply the time supervisory

# 4 EXPERIMENTAL RESULTS

The experiments described in this section have been tested in a B21 robot by RWI. All the system has been implemented in C++ language, using the system specified by CORBA which provides interoperability between objects in a heterogeneous, distributed environment.

To test the navigation system, a mission was requested to the robot. The robot started at the beginning of the corridor and it should access to C12 lab, as it is shown in figure 7.

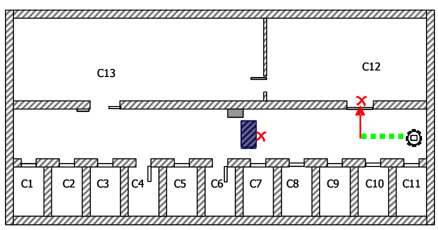


Figure 7: Experimental environment

As an experiment for the monitoring and supervision system the C12 lab door was momentarily closed when the robot was supposed to detect it. The second problem appeared in the corridor, the corridor was blocked not letting the robot continue its plan, as it is shown in figure 7.

These were the succession of steps that the system applied without any human intervention. While monitoring distance and time, the distance stored in the Navigation Chart indicated that there was a problem and the robot changed the first plan. In figure 8, it is shown the navigation interface that represents the Navigation Chart on the left and the plan obtained from the planner on the right. A failure in the node that implies detecting the door has happened.

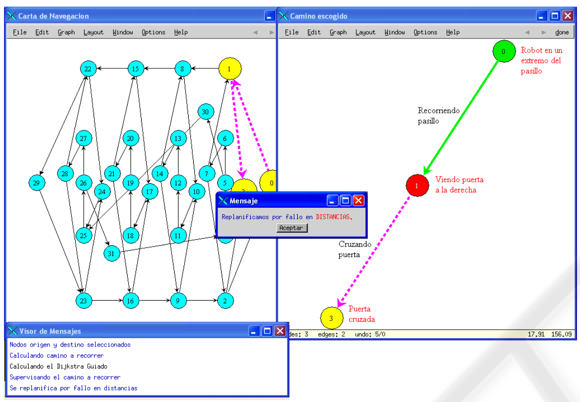

Figure 8: Navigation Chart when an error was detected

The navigation skill notified the problem and the planner generated a new plan solving the problem. The navigation skill took the new plan and began the sequence of movements, monitoring distance and time again. The new plan is shown in figure 9.

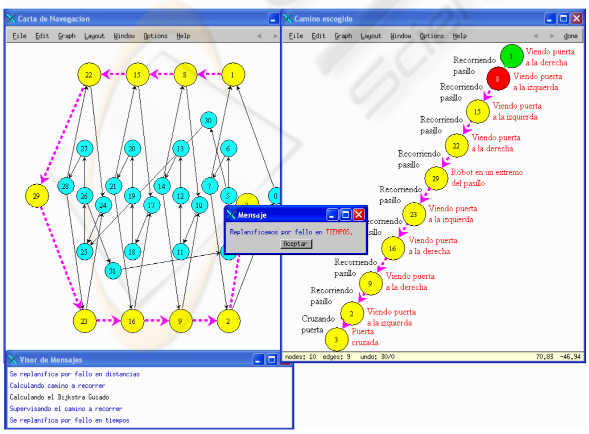

Figure 9: New plan generated and new failure detected

Then, because of the obstacle in the middle of the corridor, the time stored in the Navigation Chart indicated that there was a problem and the robot changed the plan again. In this case, the robot turned 180º placing itself in the symmetrical edge and the robot set a new plan. As the door of lab C12 was opened during the explained process then it could finally go into the lab without any problem.

## ACKNOWLEDGMENT

The authors gratefully acknowledge the funds provided by the Spanish Government through the MCYT projects TAP1999-214 and DPI2002-00188.

## REFERENCES

Alami, R.; Chatila,, R.; Fleury, S.; Ghallab, M.; Ingrand, F. An Architecture for Autonomy *International Journal of Robotics Research*, Vol. 17, Nº 4, pp 314-337, April 1998.

Barber, R.; Salichs, M.A. A new human based architecture for intelligent autonomous robots. *The Fourth IFAC Symposium on Intelligent Autonomous Vehicles*, pp85-90. 2001

Beccari, G.; Caselli, S.; Zanichelli, F.; Diemmi, D. Inducing topological maps from task-oriented perception and exploratory behaviors. *Proceedings Second EUROMICRO Workshop on Advanced Mobile Robots*. P. 134-140. IEEE Comput. Soc. USA. 1997.

Brooks, R. A robust layered control system for a mobile robot *IEEE Journal Robotics and Automation.* (1986).

Egido, V.; Barber, R.; Boada, M.J.L.; Salichs, M.A. Self-Generation by a Mobile Robot of Topological Maps of Corridors. *IEEE International Conference on Robotics and Automation*, p2662-2667.Washington, 2002.

Fernandez, J.L.; Simmons, R. Robust Execution Monitoring for Navigation Plans. .*Proceedings of Conference on Intelligent Robotics and Systems.* Vancouver Canada, October 1998.

Fichtner, M.; Grossmann, A.; Thielscher, M. Intelligent Execution Monitoring in Dynamic Environments. *Proc. Of IJCAI-03 Workshop on issues in Designing Physical Agents for Dynamic Real-Time Environments*, Acapulco, Mexico, August 2003.

Giacomo, G; Reiter, R.; Soutchanski, M. Execution monitoring of high-level robot programs. *Proc. Of the Int. Conf. On Principles of Knowledge and Representation and Reasoning*, p 453-465, 1998.

Remolina, E., Kuipers, B. Towards a General Theory of Topological Maps. *Artificial Intelligence,* 152, pp 47-104. 2004.

Stuck, E.R. Detecting and Diagnosing Mistakes*. Proc. Of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, August 1995.