The new lighting system, which can be partially seen in figure 1, is composed by six spotlights based on LED technology. Thanks to this change, the power consumption has been dramatically reduced from 400W of the original halogen lamps system to 18W without loss of lighting quality.

Another important change concerns the internal structure containing the processing unit and all the additional electronic components (see figure 4). This new structure has been built in aluminium to ensure proper heat dissipation thanks to the direct contact of one of its faces with water. Furthermore, all the heating parts of the system are in direct contact with this structure.

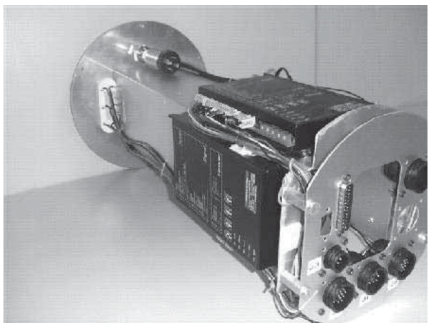

***Figure 4. Aluminium based electronics enclosure***

## 5. Power system

The power system is based on lead battery cells. The whole pack provides 48V and has a capacity of 8A/h, providing an estimate autonomy of almost 1h in normal operation. All the additional voltages needed in the vehicle are obtained from DC/DC modules.

## 6. Future work

Future work includes extensive testing in pool and open waters to adjust system and algorithm parameters and to evaluate the need for more navigation and mission sensors like a GPS and the above-mentioned IMU.

## 7. References

[1] G. Oliver, J. Antich and A.Ortiz "Pipe and cable inspection in the AIRSUB project", Instrumentation Viewpoint, 4 (1), pp 28-29, 2005.

[2] A. Ortiz, M. Simó and G. Oliver, "A vision system for an underwater cable tracker", Machine Vision and Applications, 13 (3), pp 129-140, 2002.

[3] J. Antich, A. Ortiz, "Development of the control architecture of a vision-guided underwater cable tracker", International Journal of Intelligent Systems, 20 (5), pp 477-498, 2005.

---

# REACTIVE NAVIGATION IN TROUBLESOME ENVIRONMENTS: T$^2$ STRATEGIES

Javier Antich, Alberto Ortiz

*Dep. de Matemàtiques i Informàtica, Universitat de les Illes Balears*
*Cra. Valldemossa km. 7,5 07122-Palma (SPAIN)*
*(+34) 971.172911 javier.antich@uib.es*

## 1. Introduction

Reactive mobile robot navigation based on potential field methods (PFM) [1] has shown to be a good solution for dealing with unknown and dynamic scenarios, such as, for instance, in undersea environments, where timely responses are required. Unfortunately, the complexity of the tasks which can successfully be carried out is limited by the inherent shortcomings of the approach, being the trapping situations due to local minima the most often cited [2]. This work proposes a solution to the local minima problem by introducing the traversability and tenacity (T2) principles. As a result, navigation is achieved in troublesome scenarios such as the typical U-shaped canyon or even in maze-like environments. A set of up to three variants of T2 are also put forward to ensure, whenever possible, the completion of any navigation task.

## 2. Fundamentals of the T2 Approach

The classic potential field approach [1] is the basic framework for the application of the T2 family of navigation strategies. It computes the motion of the robot on the basis of two simple behaviours: GoTo and AvoidObstacles. More precisely, the former generates an attractive force in direction to the goal, while the latter considers obstacles as repulsive surfaces. The robot follows the negative gradient of the resulting potential field towards its minimum, whose position coincides with the goal point.

The inability to move the robot away from the goal direction in a non-momentary and strategic way is the main cause of the undesirable trapping situations suffered by PFM. Figure 1(a) shows an example where a robot controlled by PFM is unable to escape from a U-shaped obstacle. A solution to this problem is given next by applying the T2 principles in the context of the so-called navigation filter.

The main function of the navigation filter is the appropriate alteration of the direction of the motion vector generated by the GoTo behaviour in order to overcome any obstacle irrespective of its size and shape (figure 1(b)). Such change is carried out according to the traversability and tenacity principles.

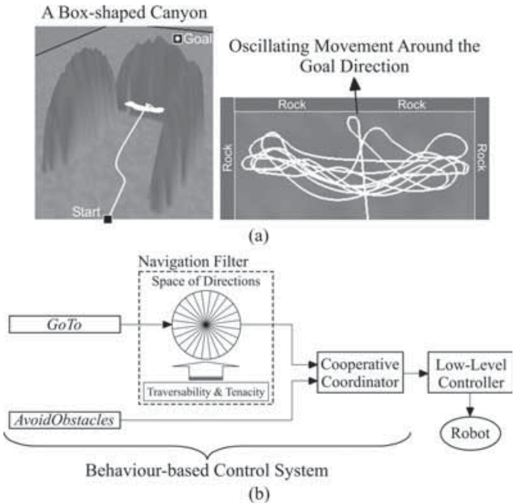

***Figure 1. (a) A typical trapping situation for PFM-based control systems. (b) Integration of the navigation filter into the classic potential field approach.***

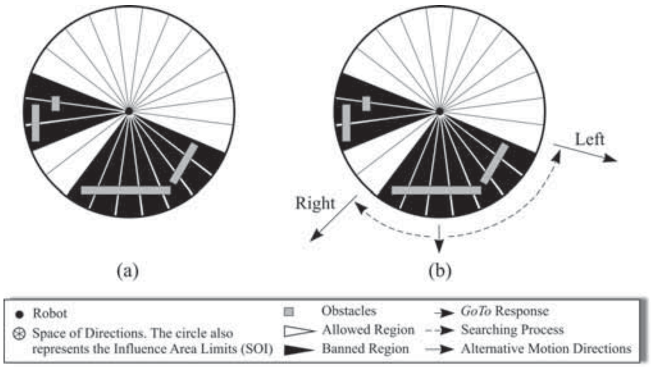

***Figure 2. (a) Division of the space of directions into regions, labelling them as allowed or banned. (b) Alternative obstacle-free motion directions.***

The traversability principle suggests banning those directions where an obstacle has been detected (figure 2(a)) and, in case the GoTo behaviour response results banned, generates two alternative obstacle-free directions, generically labelled as left and right (figure 2(b)). On the other hand, which direction, left or right, is taken by the robot is decided according to the tenacity principle, in the sense that it tenaciously suggests taking the same direction that was selected the last time. These two principles, although simple in concept, have proven fully effective to solve trapping situations. As a consequence of the application of the T2 principles, a global behaviour emerges by which the robot circumnavigates the contour of an obstacle once it is aware of its existence.

### 3. Three T2-based Algorithms

The application of the T2 principles require making two decisions during the navigation: (1) in which direction the contour of an obstacle is followed (i.e. which direction is taken the first time the tenacity principle is applied), and (2) at which moment the robot decides to leave the contour of an obstacle to continue with the navigation towards the goal point.

Random T2, Connectivity T2 and Bug-based T2 are three strategies deriving from T2 which make the two aforementioned decisions in different ways, giving rise to different performances and capabilities. They are briefly described in the following:

- In Random T2, the robot chooses randomly the direction to follow the contour, while the contour is left as soon as the direction towards the goal point becomes allowed.
- Connectivity T2 decides the direction to follow the contour according to a minimum turn criterion and remembers which direction is taken, in order to explore the other direction in case the goal point is not reached. The contour of the obstacle is also left as soon as the direction towards the goal point becomes allowed.
- Bug-based T2 also chooses the minimum turn direction to follow the contour. As for the leaving to occur, either: (1) the navigation filter must indicate there is a free-obstacle path towards the goal point and this must be the first time the robot leaves the obstacle at approximately that same position, or (2) the robot trajectory must cut the line joining the start and goal points (M-line) and the distance from the robot to the goal has to be shorter than the one associated with the last time this condition was satisfied. Each time (1) is satisfied the M-line is redefined using the current robot position as the start position.

### 4. Experimental Results

In order to show the effectiveness of the T2 strategies, figure 3 shows simulation results for a maze-like environment, while table I provides a comparison of path lengths with a well-known motion planning algorithm called Bug2 [3] for a set of up to seven different environments.

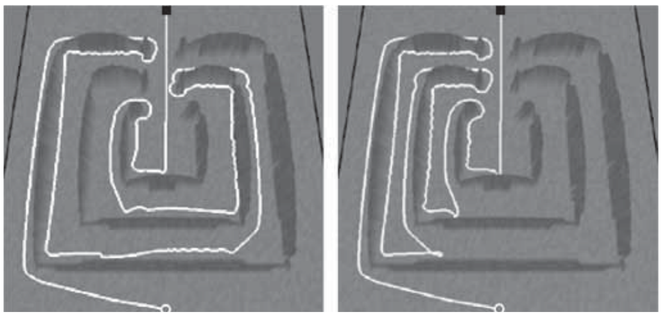

*Figure 3. Results for a troublesome mission: (a) Random T2 (worst result), (b) Connectivity and Bug-based T2 (black dot = start, red dot = goal).*

| Mission | RT2 | CT2/BT2 | Bug2 |
|---|---|---|---|
| 1 | 313.62 | 281.54 | 388.00 |
| 2 | 420.65 | 420.65 | 426.00 |
| 3 | 623.00 | 405.17 | 578.00 |
| 4 | 286.83 | 368.55 | 421.00 |
| 5 | 883.47 | 883.47 | 934.00 |
| 6 | 1586.71 | 1484.91 | 1672.00 |
| 7 | 663.82 | 246.85 | 797.00 |
| Total (m) | 4778.10 | 4091.19 | 5216.00 |

*Table I. Comparison of the path lengths of Random T2 (RT2), Connectivity T2 (CT2), Bug-based T2 (BT2) and Bug2.*

### 5. Conclusions

A novel family of geometric algorithms of motion planning based on potential fields and the traversability and tenacity principles have been put forward. The T2 variant paths have resulted, on average, significantly shorter than the ones of other algorithms for a representative set of missions. An extension of T2 strategies to three dimensions is under development at the moment to take advantage of the 6 DOF of AUVs.

### 6. References

[1] O. Khatib. "Real-time obstacle avoidance for manipulators and mobile robots," International Journal of Robotics Research, vol. 5, no. 1, pp. 90–98, 1986.
[2] Y. Koren and J. Borenstein. "Potential field methods and their inherent limitations for mobile robot navigation," in Proc. 1991 Robotics and Automation Conf., pp. 1398–1404.
[3] V. Lumelsky and A. Stepanov. "Path planning strategies for a point mobile automaton moving amidst unknown obstacles of arbitrary shape," Algorithmica, vol. 2, pp. 403–430, 1987.

---

# DEFINING A MISSION CONTROL LANGUAGE

Narcis Palomeras, Pere Ridao and Marc Carreras

*Univ. de Girona, Campus de Montilivi, Edifici P4 – Girona*
*npalomer@eia.udg.es*

### 1. Introduction:

A mission controller is the part of the control architecture that is in charge of defining a thread of tasks to be carried out in order to fulfil a mission. Each task can be executed by means of some vehicle primitives often referred as basic robot commands or behaviours. The mission controller must define how the mission is divided into a set of tasks and how primitives are combined to fulfil each task. The development of a Mission Control System (MCS) for an Autonomous Underwater Vehicle (AUV) lies at the intersection of a Discrete Event System (DES), in charge of enabling/disabling basic actions when some events are produced, and the Dynamic Control Continuous System (DCS), used for every action to achieve a specific goal [1]. The main objective of this project is defining a language able to describe AUV missions in a simple but versatile way.

The Petri net formalism [2] has been chosen to describe the missions that the MCS is going to execute, but instead of using graphic tools to describe these Petri nets, our approach uses a Mission Control Language (MCL), which compiles into a Petri net. The adoption of this formalism helps us evaluating the net properties at compilation time to detect, for instance, if possible inconsistent states can be reached. Hence corrective actions can be adopted before actual execution.

This work is structured as follows, section 2 describes our proposal