



## **A PFM-BASED CONTROL ARCHITECTURE FOR A VISUALLY GUIDED UNDERWATER CABLE TRACKER TO ACHIEVE NAVIGATION IN TROUBLESOME SCENARIOS**

J. Antich<sup>1</sup>, A. Ortiz<sup>2</sup> and G. Oliver<sup>3</sup>

### **ABSTRACT**

Nowadays, the surveillance and inspection of underwater installations, such as power or telecommunication cables and pipelines, are carried out by trained operators who, from the surface, control a Remotely Operated Vehicle (ROV) with cameras mounted over it. This is a tedious, time-consuming and expensive task, prone to errors mainly because of loss of attention or fatigue of the operator and also due to the typical low quality of seabed images. In this study, a control architecture based on Potential Field Methods (PFM) for visually guiding an Autonomous Underwater Vehicle (AUV) to detect and track a cable, or pipeline, laid on the seabed is presented. Additionally, a solution to the typical trapping problem linked to this kind of control systems is proposed. The efficiency of the solution is evaluated and compared against other popular strategies appearing in the literature. A 3D simulation environment which incorporates the hydrodynamic model of a real underwater vehicle called GARBI has been used with this purpose.

**Key words:** Autonomous Underwater Vehicles, Obstacle Avoidance, Local Path Planning, Vision-based Pipeline Tracking.

---

<sup>1</sup> Profesor Ayudante de Escuela Universitaria, Universidad de las Islas Baleares, (javi.antich@uib.es), Telf. 971172911, Fax 971173003, Escuela Politécnica Superior. Ctra. Valldemossa km 7.5, 07071 Palma de Mallorca, Spain.

<sup>2</sup> Profesor Titular de Escuela Universitaria, Universidad de las Islas Baleares (alberto.ortiz@uib.es) Telf. 971172992, Fax 971173003, Escuela Politécnica Superior. Ctra. Valldemossa km 7.5, 07071 Palma de Mallorca, Spain.

<sup>3</sup> Profesor Titular de Universidad, Universidad de les Illes Balears (goliver@uib.es), Telf. 971173201, Fax 971173003, Escuela Politécnica Superior. Ctra. Valldemossa km 7.5, 07071 Palma de Mallorca, Spain.



## INTRODUCTION

The feasibility of an underwater installation can only be guaranteed by means of a suitable inspection programme. This programme must provide the company with information about potential hazardous situations or damages caused by the mobility of the seabed, corrosion, or human activities such as marine traffic or fishing. Nowadays, the surveillance and inspection of these installations are carried out using video cameras attached to ROVs normally controlled by operators from a support ship. Obviously, this is a tedious task because the operator has to concentrate for a long time in front of a console, which makes the task highly prone to errors mainly due to loss of attention and fatigue. Besides, undersea images possess some peculiar characteristics which increase the complexity of the operation: blurring, low contrast, non-uniform illumination and lack of stability due to the motion of the vehicle, just to cite some of them. Therefore, the automation of any part of this process can constitute an important improvement in the maintenance of such installations with regard to errors, time and monetary costs.

To this end, an automatic system for autonomously locating and tracking a cable/pipeline on the basis of visual feedback is presented and discussed in this paper. For a start, the rigidity and shape of the underwater cable is exploited by a computer vision algorithm to discriminate the cable from the surrounding environment even in very noisy images. A suitable control architecture makes then use of the output of the vision system to make the AUV track the cable. Behaviours and PFM are in the heart of the proposed architecture. Besides, it integrates a novel reactive obstacle avoidance/navigation strategy to escape from both local and global trapping situations. A 3D OpenGL-based simulator incorporating the dynamics model of a real underwater vehicle has been used to validate the proposal.

The rest of the paper is organized as follows: the vision system is briefly described in section 2; section 3 presents the control architecture, while the navigational approach is discussed in section 4; and, finally, the last section gives some conclusions and future work.

## THE VISION SYSTEM

### Brief Description

Artificial objects usually present distinguishing features in natural environments. In the case of the cable, given its rigidity and shape, strong alignments of contour pixels can be expected near its sides. The vision system exploits this fact to find the cable in the images.

To this end, every image is first segmented by means of a classification algorithm working over an (intensity, intensity gradient) space. Next, from the region contours, alignments of contour pixels are determined and analysed. If among those alignments there is strong evidence of the location of the cable (mainly two align-



ments with a great number of pixels lined up and with a high degree of parallelism, even without discounting the perspective effect), then the cable is considered to have been located and its parameters are computed. Otherwise, the image is discarded and the next one is analysed.

Once the cable has been detected, its location and orientation in the next image are predicted by means of a Kalman filter, which allows reducing the pixels to be processed to a small ROI (Region Of Interest). In this way, the computation time is considerably lowered together with the probability of misinterpretations of similar features appearing in the image.

When tracking the cable, a low or null evidence of its presence in the ROI can be obtained. In such a case, a transient failure counter is increased after discarding the image. If this anomalous situation continues throughout too many images, then it is attributed to a failure in the prediction of the ROI, resulting in two special actions: the Kalman filter is reset and the ROI is widened to the whole image.

Detailed information about the different steps of the system can be found in (Antich and Ortiz, 2003).

### Presentation of Some Results

The vision system has been tested using sequences from a video tape obtained in several tracking sessions of various real cables with an ROV driven from the surface. These cables were installed several years ago, so that the images do not present highly contrasted cables over a sandy seabed; on the contrary, these cables are partially covered in algae or sand, and are surrounded by algae and rocks, making thus the sequences highly realistic. The mean success rate that has been achieved is about 90% for a frame rate of more than 40 frames/second on average.

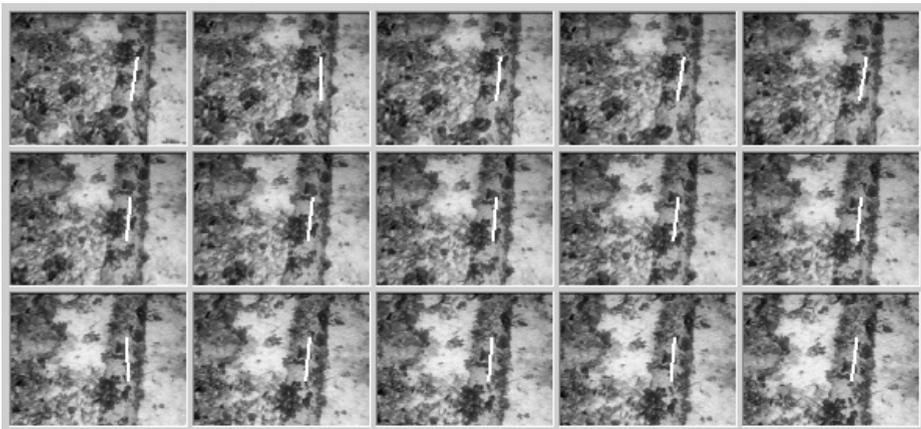


Figure 1 shows results for an excerpt of one of the testing sequences.

## A BEHAVIOUR-BASED CONTROL ARCHITECTURE

### Framework

Robot control is the process of taking information about the environment, through the sensors of the robot, processing it as necessary to decide how to act, and then executing those actions by means of the available effectors to achieve the set of goals corresponding to a user-specified mission. Nowadays, there is a small number of fundamentally different classes of robot control methodologies (see (Ridao et al., 1999) (Valavanis et al., 1997), among many others) which are usually embodied in particular control architectures and that can be roughly classified as: deliberative/hierarchical, behavioural/reactive, and hybrid. However, only the two last ones are suitable to deal with complex, non-structured, and changing worlds, like the majority of underwater environments. This paper proposes a complete behavioural control architecture based on Motor Schemas with 3D Potential Fields to control the navigation of the underwater cable tracker. Simplicity and modularity have been the key factors to choose this approach.

In this context, behaviours are the basic building blocks to carry out robotic actions, representing each of them the reaction to a stimulus. Behavioural responses are all coded as 3D vectors whose orientation denotes the direction to be followed by the vehicle while the magnitude expresses the strength of the response against other behavioural commands. The vectors generated by the architecture's active behaviours are asynchronously channelled into a coordination mechanism that combines them in order to obtain the final control system response.

### Mission Stages and Sensory Equipment

Several stages are distinguished within a cable-tracking mission: diving, sweeping, tracking and homing. In the first one, the AUV, after having been released from the support ship, goes down until a certain distance to the seabed is reached. The second and third stages comprise, respectively, searching for the cable in a predefined exploration area and tracking it once found. Finally, the vehicle returns to the starting point after having achieved the limits of the exploration area while tracking the cable.

The control architecture assumes the existence of three different kinds of sensors to carry out cable-tracking missions: forward and side-looking sonars, a compass and a camera. Furthermore, the position of the vehicle is estimated by means of an acoustic positioning system of the so-called Long Base Line (LBL) type.

### Behaviour Description

Taking into account the aforementioned general way of action for the AUV, the reactive control layer of the vehicle was split into five primitive behaviours.



Some of them appear in the classical literature about behavioural architectures, but others are specific of this application. They all are described in the following:

- *Stay on region* prevents the AUV from straying from the area to be explored. The behaviour is exclusively activated when the vehicle is close to the limits of the exploration area. In such a case, a vector that moves the vehicle away from those limits is generated, being its magnitude directly related to the corresponding distance: the closer to the limits, the larger the magnitude.
- *Avoid obstacles* allows the vehicle to avoid navigational barriers such as rocks, algae or, even, other possible cooperating vehicles. In this case, a vector in the opposite direction to the obstacles is generated. The magnitude of the vector is again variable, now according to the distance that separates the AUV from the obstacles ahead.
- *Cable detection and tracking* moves the vehicle strategically through the exploration area in search of a sufficient evidence of the presence of the cable. Specifically, after having acquired the working depth through a vertical path from the surface, the AUV executes the sweeping stage performing a zigzag movement on the exploration area until the cable is found. Although other more optimised strategies could have been devised, it is important to notice that it has been assumed a total lack of information about the location of the cable, so that there are no many more alternatives but an exhaustive or near-exhaustive search.

Once the cable has been detected, the tracking stage starts. At this point, the AUV can be oriented in any one of the two possible—and opposite— directions to start tracking the cable. The particular choice is based on a predefined parameter which establishes a certain range of preferred orientations.

Along the tracking, two different tasks are sequentially executed: the first one tries to keep the cable oriented vertically in the field of view (FOV), while the second task intends to maintain the cable in the central area of the FOV. In this way, improvements in both the cable visual detection and the longitude and smoothness of the vehicle's path are expected.

As can be easily anticipated, anomalous situations can arise in a real application. In particular, the cable can disappear from the images because the AUV's course has drifted apart from the actual cable location. In such cases, a suitable recovery mechanism is activated, consisting in making the behaviour return to its internal search state, where the vehicle acquires again the zigzag movement. However, now the area to be explored is reduced using the vehicle's trajectory during the past tracking stage (see figure 2). This trajectory is fitted by a straight line and a new search zone is determined computing the intersection between such line and the limits of the exploration area. Note that the dimensions of the new exploration area can be readjusted according to the AUV's manoeuvrability.

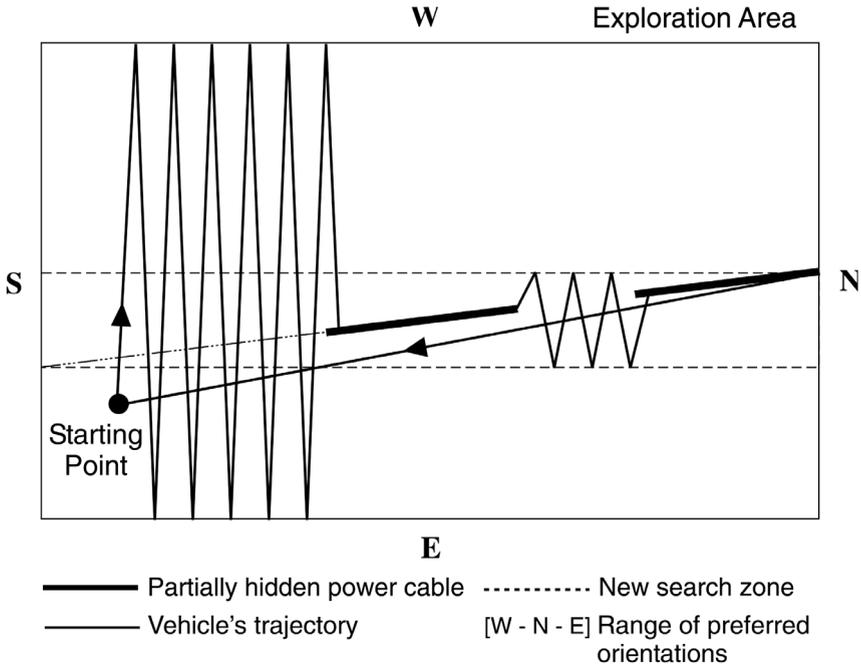


Figure 2. AUV's trajectory for a typical mission

- *Keep distance to seabed* tries to maintain the distance to the seabed constant in order to keep the apparent width of the cable in the images also constant. In this way, the vision subsystem can assume that the separation between both sides of the cable does not vary, and use this information to reduce its probability of failure. Sonars or, in case they cannot bring accurate enough measures, the acoustic positioning system, are expected to supply the required distance to the seabed.
- *Go home*, finally, makes the vehicle go to the mission start point. Two different steps are carried out: first, the AUV approaches to the starting point keeping a certain distance to the seabed; afterwards, it goes up until the sea surface is reached. In both cases, the magnitude of the output vector is proportional to the proximity to the intermediate/final goals considered.

### A Hybrid Coordination Mechanism

As it has been said, the output vectors generated by the architecture's active behaviours are asynchronously channelled into a coordination mechanism which is responsible for obtaining the final control system response. Such mechanisms are typically classified into two groups: competitive or cooperative. In the former, the control of the robot is given to one behaviour until the next execution cycle. On the



contrary, the latter combines recommendations from multiple behaviours to form a single control action which represents their consensus. Both strategies offer advantages and disadvantages. On the one hand, competitive methods provide good robustness but non-optimal paths from the point of view of their smoothness and length. Cooperative methods, on the other hand, present the opposite features.

After the previous analysis, it seems obvious that a hybrid methodology that is able to make the most of both coordination strategies is desirable. With this aim, a new concept called *activity level* has been introduced. It represents the urgency of a behaviour for taking the control of the robot in a certain moment, expressed by a real value within the interval [0,1]. Additionally, the hybrid coordinator assigns, in a dynamic way, priorities low and high to the behaviours according to whether they generate an activity level above or below a user-defined value. Moreover, a different threshold can be established for each behaviour. Once the prioritisation process has finished, the coordinator acts competitively between the aforementioned priority levels and cooperatively inside them.

Some comparative data between the cooperative and the hybrid approaches, obtained by simulation, are provided in figure 3. More precisely, the figure shows the resistance of a vehicle to collide when it is trapped into a box-shaped canyon and the goal point is beyond the walls. During the experiment, the gain of a GoTo-type behaviour was progressively increased keeping constant the weight of the basic obstacle avoidance primitive, whose activity level, for the hybrid case, was inversely proportional to the distance to the nearest obstacle detected. As can be observed in the figure, the hybrid coordination mechanism managed to avoid collisions for the length of all the simulations, while the cooperative version led to a collision almost all the times.

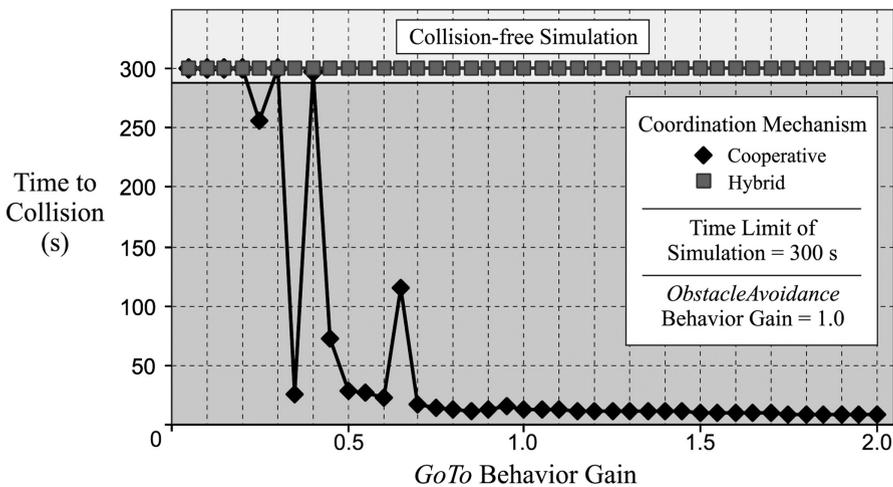


Figure 3. A comparison between a cooperative and a hybrid coordination mechanism

## Identified Shortcomings

In robotic navigation, potential field methods are a well-known solution for dealing with unknown and dynamic scenarios by taking into account the reality of the environment during the robot motion. The characteristic elegance and simplicity of the approach when representing and successfully solving a path-planning problem in real-time explains its extensive application in this field. However, substantial shortcomings have been identified as problems inherent to this principle (Koren and Borenstein, 1991). *Getting stuck in local minima* is the best-known and most often-cited problem with PFMs. Consequently, several obstacle configurations may lead to undesirable trapping situations. Figure 4 shows, by way of example, how the underwater cable tracker is unable to escape from a U-shaped canyon. This result was obtained by using a 3D simulation environment named NEMO<sub>CAT</sub> (Antich and Ortiz, 2004 (a)) which incorporates the dynamic model of a real underwater vehicle called GARBI, designed and built by the Computer Vision and Robotics research group of the University of Girona (Spain), making thus the simulations more realistic.

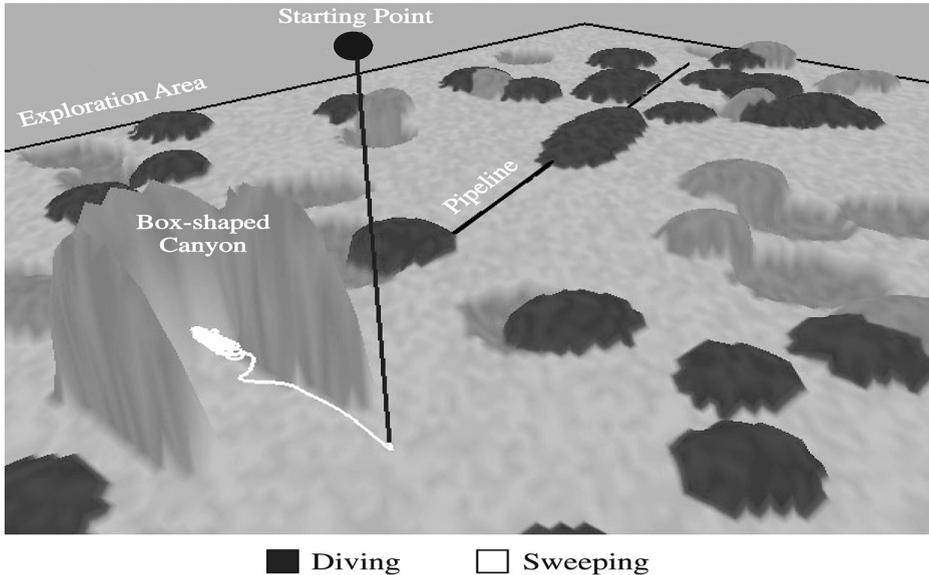


Figure 4. The AUV, after the diving stage, is trapped into a box-shaped canyon

In the next section, a new component will be added to the control architecture in order to solve the aforementioned problem.



## A NEW NAVIGATIONAL APPROACH

A navigation strategy called *Random T<sup>2</sup>* will be explained now. It gives an efficient solution to the trapping problem by applying two new concepts: *Traversability* and *Tenacity*. Figure 5 shows the new navigation module integrated in the framework of the previously described behaviour-based control architecture. It mainly consists of a filtering process which makes the robot take a different direction to the one suggested by the coordination mechanism in case it is not appropriate for the present configuration of surrounding obstacles. The details are given in the following, together with a comparative study on the path length performance of the proposal for a series of experiments.

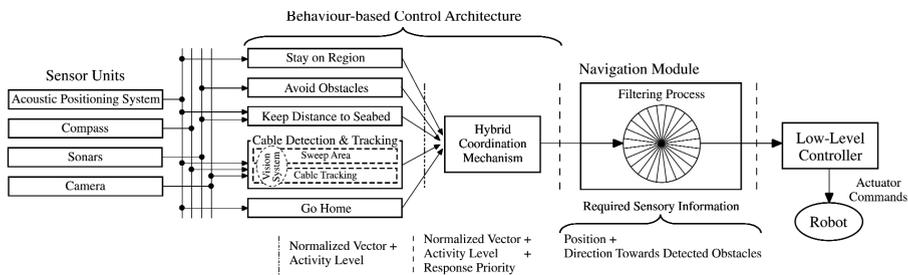


Figure 5. Complete underwater cable tracker's control system

### The Filtering Process

The appropriate alteration of the direction of the motion vector generated by the coordination mechanism pertaining to the control architecture is the main concern of the filtering process. Such change is carried out according to the *traversability* and *tenacity* concepts. The former suggests banning those directions where an obstacle has been detected, choosing an obstacle-free direction close to the desired one. The latter, on the other hand, determines the way how that selection process has to be done. Essentially, a criterion of avoiding abrupt changes in the robot heading is applied.

As can be observed in figure 6(a), the space of directions around the robot is divided into  $K$  identical angular regions, and obstacles detection information for each region is stored in a suitable data structure, which also registers the approximate obstacles location. This information has to be kept updated as the robot moves, even when no new obstacles are detected in the environment (see figure 6(b)). Finally, in this context, a region is said to be *banned* when at least one obstacle is known to be in the range of directions which consists of.

After receiving the output of the reactive control system, its viability is studied on the basis of the traversability principle. Changes are required only if the

direction of the motion vector belongs to a banned region. In such a case, two alternative directions are obtained (see figure 6(c)) and a decision has to be made. In this work, this decision is taken in a random way the first time. Afterwards, the tenacity principle is applied by choosing between the two options the region closest to the one taken the last time.

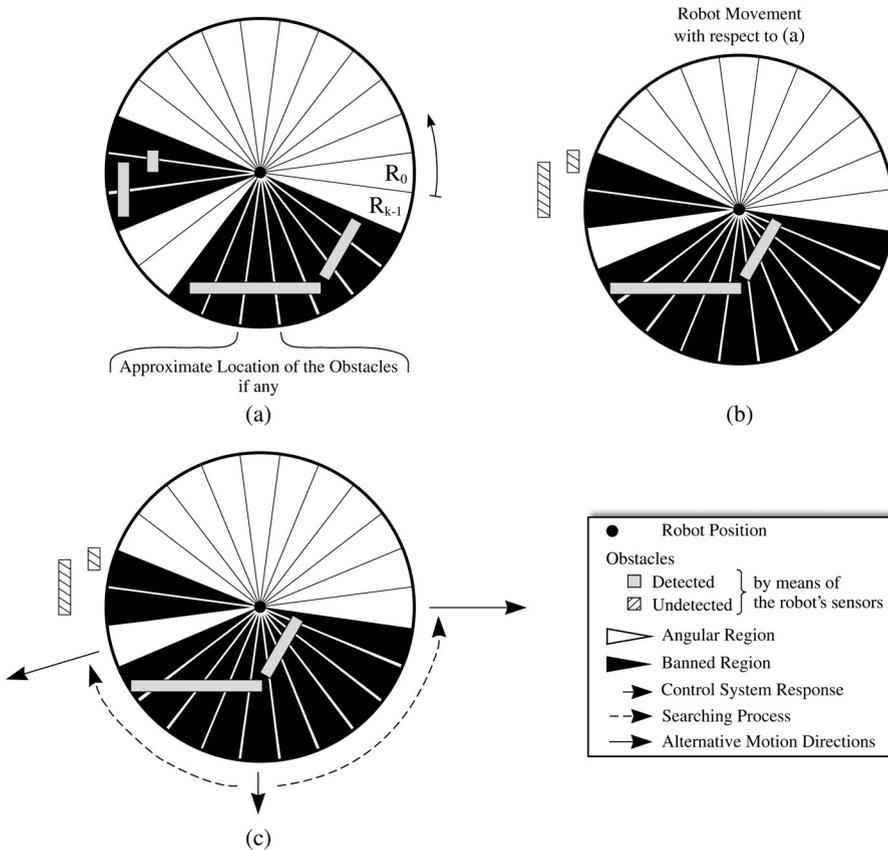


Figure 6. Implementation of the traversability concept: (a) division of the space of directions around the robot; (b) maintaining the region data coherence while navigating; (c) selection of two apparent obstacle-free motion directions

The application of all these steps results in a global system behaviour that can be summarized in two points:

- When the robot is navigating far from obstacles, it heads for its current goal which will be defined by either the *cable detection and tracking* or the *go home* behaviours depending on the mission stage. During this period of time, the filtering process remains inactive.



— After the detection of an obstacle, on the other hand, the robot follows its contour in a certain randomly-chosen direction until a way out is found. The repeated input of motion vectors linked to non-banned regions indicates the end of a presumed trapping situation. In such a case, the filter is reset losing thus all the previously kept information.

By way of example, figure 7 shows how a robot is able to escape from a U-shaped canyon by applying the described process.

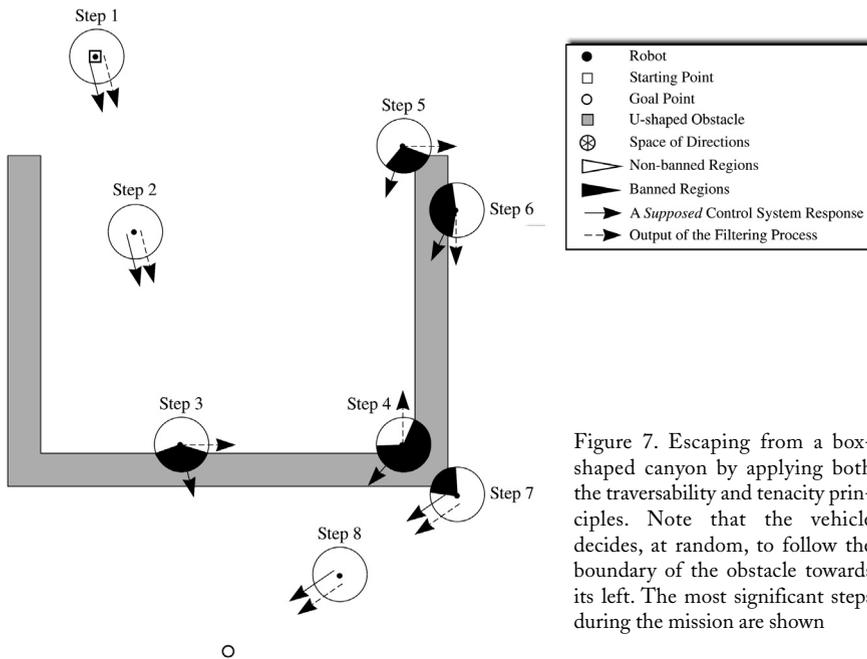


Figure 7. Escaping from a box-shaped canyon by applying both the traversability and tenacity principles. Note that the vehicle decides, at random, to follow the boundary of the obstacle towards its left. The most significant steps during the mission are shown

It is important to note how relatively easy this strategy detects potentially deadlock situations, mainly due to environment concavities, which may result in a cyclic behaviour. These are characterized by a space of directions fully, or almost fully, banned. In this kind of cases, contrary to other approaches, the robot tenaciously continues following the obstacle boundary rejecting an apparently correct but, in general, wrong path towards the goal.

In this way, the proposed navigation module solves the best-known and most often-cited problem of PFMs: trapping situations due to local minima. However, some other interesting properties stem from the application of the aforementioned principles. On the one hand, oscillations in both the presence of obstacles and in narrow corridors are significantly reduced owing to the restricted motion of the robot in directions where obstacles have not been detected. Better stability results are obtained when the robot is, additionally, forced to follow the obstacle contour in

only one direction by filtering sudden robot heading changes suggested by the reactive control layer. As a consequence, these robots are certainly suitable to navigate in cluttered environments with closely spaced obstacles.

### A Comparative Study

In the following, a comparative study on the performance of the navigation strategy proposed is presented and discussed. The comparison is carried out from the point of view of a single criterion which is the length of the path between the starting and the goal points. The other algorithms considered in the study are: *Avoiding the Past*, *Learning Momentum* (LM), and *Micronavigation* (mNAV). A brief description of each of them is given next:

- *Avoiding the Past* makes the robot avoid recently visited locations by maintaining and utilizing a local spatial memory (Balch and Arkin, 1993).
- *LM* adjusts the behavioural parameters of a reactive control system at runtime depending on which one of several predefined situations the robot is in (Lee and Arkin, 2001).
- *μNAV* is a PFM-based approach which provides the robot with a hierarchy of simple behaviours designed for smooth obstacle avoidance and for escaping from concavities (Scalzo et al., 2003).

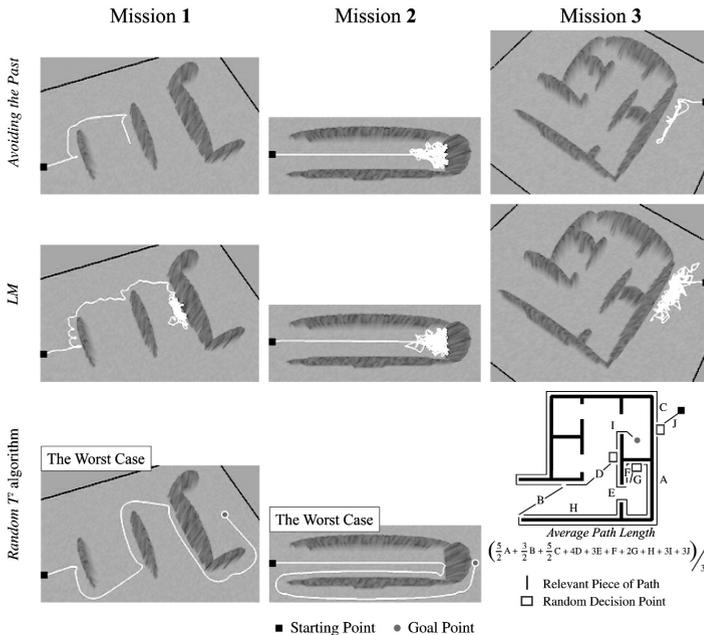


Figure 8. From top to bottom, paths generated by the Avoiding the Past, LM, and Random T2 approaches in NEMOCAT. Note that due to the non-deterministic behaviour of our strategy, different results can be obtained in different runs of the simulator. Only one, the worst, is shown in the figure whenever possible. A stochastic analysis about the average length of the AUV's path is presented when such worst case cannot be computed due to its infinite extent (see the details about this in (Antich and Ortiz, 2004 (b))



NEMO<sub>CAT</sub> was used again to measure the path length of the two first navigational approaches as well as *Random T<sup>2</sup>* under a simplified framework where a behaviour of the control architecture, *cable detection and tracking* or *go home*, was supposed to establish the only goal point considered for each mission. As can be observed in figure 8, three environments were defined. In the first one, walls/rocks of different length impede the progress of the vehicle towards its goal. The second environment, on the other hand, corresponds to a very deep box-shaped canyon. Finally, the third one appeared in (Ranganathan and Koenig, 2003), where a control system with deliberative capabilities was employed to solve it. *Avoiding the Past* and *LM* strategies were not able to successfully carry out any of the previously described missions, which shows their poor effectiveness to escape from large trapping areas. In both cases, the simulation was stopped after a travel time twice that of *Random T<sup>2</sup>*.

In order to continue with the study, a robot programming environment based on the AuRA (Arkin and Balch, 1997) architecture called *MissionLab* (Mackenzie et al., 1997) was also used. The latest release of this software (version 6.0) integrates the *mNAV* algorithm implemented by one of its authors. Different tests with increasing complexity were performed in *MissionLab*, simulating a holonomic robot equipped with several range finders, and wheel encoders to compute its position by means of dead-reckoning. As can be verified in (Scalzo et al., 2003), such experiments are a representative sample of the whole power of the *mNAV* strategy for a typical behaviour hierarchy. Each environment was then accurately reproduced in NEMO<sub>T</sub> and successfully solved by *Random T<sup>2</sup>*. The results obtained are shown in figure 9. Besides, table 1 compares the performance of both strategies from the viewpoint of the resultant path length.

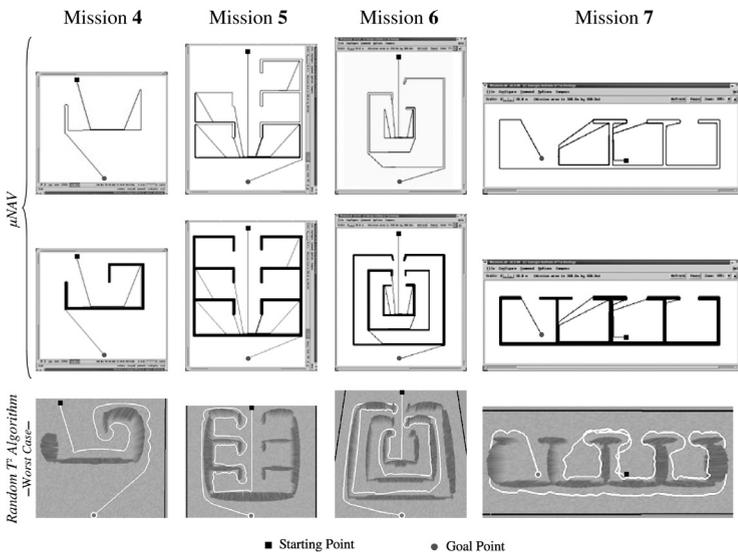


Figure 9. Simulation results for the  $\mu NAV$  (top -walls removed- and middle) and *Random T<sup>2</sup>* (bottom) algorithms by using *MissionLab* and NEMO<sub>CAT</sub>, respectively

Mission	Algorithm Type		Improvement (%)	$\frac{\mu NAV}{Random T^2}$
	Random $T^2$ (average <sup>1</sup> )	$\mu NAV$		
4	286.83	384.20	25.34	1.34
5	883.47	4180.41	78.87	4.73
6	1586.71	2489.40	36.26	1.57
7	675.42	1316.40	48.69	1.95
Total	3432.43	8370.41	58.99	2.40 (average)

Table 1. Comparing the path lengths (m) of Random  $T^2$  and the  $\mu NAV$  strategies

As can be observed, our proposal produced, on average, trajectories between the starting and goal points 2.4 times shorter than  $\mu NAV$ . The difference derives from the fact that  $\mu NAV$  allows the robot, in general, to head for the goal as soon as it is faced without any immediate obstacle in the way, while *Random  $T^2$*  limits the applicability of such rule to situations where a concavity is not detected.

The reader is referred to (Antich and Ortiz, 2004 (b)) for a more extensive comparative study using both simulation and real results. Specifically, a representative member of the popular Bug family of robot motion planning algorithms called *Bug2* is, in addition, taken into account.

## CONCLUSIONS AND FUTURE WORK

A behaviour-based control architecture for visually guiding an AUV to detect and track a cable, or pipeline, laid on the seabed has been presented. A novel strategy to avoid the typical trapping problem associated with this kind of control systems has been proposed as the main contribution to this work. The approach called *Random  $T^2$*  provides a solution in both local and global level by using a reduced amount of memory and computational resources. *Random  $T^2$*  has been compared against other well-known algorithms sharing the same goal (*Avoiding the Past*, *Learning Momentum*, and *Micronavigation*). The length of the resulting paths has been used as the figure of merit. Our proposal generated, on average, trajectories 2.4 times shorter. These results were obtained by means of an underwater simulation environment named NEMO<sub>CAT</sub>. In the near future, experimentation at sea with our SeaLion robot (see figure 10) will be carried out. An extension to three dimensions will also be addressed in the future to take advantage of the larger number of DOFs of underwater robots against land robots.

<sup>1</sup> In order to compute this value, all the possible paths between the starting and the goal points together with their corresponding probabilities of happening have been considered for each mission.



Figure 10. Torpedo-like robot belonging to our research group SRV

## ACKNOWLEDGEMENTS

This study has been partially supported by project CICYT-DPI2001-2311-C03-02 and FEDER funds. Its authors wish to thank the Georgia Tech Research Corporation for the free distribution of the software *MissionLab*.

## REFERENCES

- Antich, J. and Ortiz, A. (2003) Underwater Cable Tracking by Visual Feedback, *Proceedings of the 1st Iberian conference on Pattern Recognition and Image Analysis*, 4-6 June, Mallorca, Spain, pp. 53-61.
- Antich, J. and Ortiz, A. (2004 (a)) An Underwater Simulation Environment for Testing Autonomous Robot Control Architectures, *Proceedings of the IFAC conference on Control Applications in Marine Systems*, 7-9 July, Ancona, Italy, pp. 509-514.
- Antich, J. and Ortiz, A. (2004 (b)) *T<sup>2</sup>: An Approach to Robotic Navigation in Unknown and Dynamic Environments*. Technical Report A-3-2004. University of the Balearic Islands, Spain.
- Arkin, R. and Balch, T. (1997) AuRA: Principles and Practice in Review. *Journal of Experimental and Theoretical Artificial Intelligence* 9 (2), 175-189.
- Balch, T. and Arkin, R. (1993) Avoiding the Past: a Simple but Effective Strategy for Reactive Navigation, *Proceedings of the International Conference on Robotics and Automation*, vol. 1, pp. 678-685.
- Koren, Y. and Borenstein, J. (1991) Potential Field Methods and their Inherent Limitations for Mobile Robot Navigation, *Proceedings of the International Conference on Robotics and Automation*, vol. 2, pp. 1398-1404.



- Lee, J. and Arkin, R. (2001) Learning Momentum: Integration and Experimentation, *Proceedings of the International Conference on Robotics and Automation*, vol. 2, pp. 1975-1980.
- Mackenzie, D. and Arkin, R. and Cameron, J. (1997) Multiagent Mission Specification and Execution. *Autonomous Robots 4 (1)*, 29-52.
- Ranganathan, A. and Koenig, S. (2003) A Reactive Robot Architecture with Planning on Demand, *Proceedings of the International Conference on Intelligent Robots and Systems*, vol. 2, pp. 1462-1468.
- Ridao, P. and Batlle, J. and Amat, J. and Roberts, G. N. (1999) Recent Trends in Control Architectures for Autonomous Underwater Vehicles. *International Journal of Systems Science 30 (9)*, 1033-1056.
- Scalzo, A. and Sgorbissa, A. and Zaccaria, R. (2003) mNAV: a Minimalist Approach to Navigation, *Proceedings of the International Conference on Robotics and Automation*, vol. 2, pp. 2018?-2023.
- Valavanis, K. and Gracanin, D. and Matijasevic, M. and Kolluru, R. and Demetriou, G. (1997) Control Architectures for Autonomous Underwater Vehicles. *IEEE Control Systems Magazine 17*, 48-64.



## UN SISTEMA DE CONTROL REACTIVO BASADO EN CAMPOS DE POTENCIAL PARA UN VEHÍCULO CAPAZ DE DETECTAR Y SEGUIR VISUALMENTE CABLES SUBMARINOS EN ESCENARIOS COMPLEJOS

La operabilidad de una instalación submarina  $\frac{3}{4}$  cables de transporte de energía eléctrica o de telecomunicaciones y/o tuberías  $\frac{3}{4}$  puede sólo ser garantizada a través de un programa de inspección capaz de proporcionar a tiempo información sobre condiciones de peligro potenciales o daños causados por la movilidad del suelo oceánico, la corrosión o actividades humanas tales como el tráfico marino y la pesca. Hoy en día, estas tareas de vigilancia e inspección son realizadas por operadores que desde la superficie de un barco controlan un ROV (Remotely Operated Vehicle) sobre el que se han montado cámaras de vídeo. Evidentemente, ésta es una tarea tediosa en la que el operador debe permanecer largos periodos de tiempo concentrado en frente de una consola, favoreciendo todo ello la aparición de errores cuyo origen es, principalmente, la pérdida de atención y la fatiga. Además, las peculiares características de las imágenes obtenidas del fondo marino  $\frac{3}{4}$  *blurring*, bajo contraste, iluminación no uniforme, etc.  $\frac{3}{4}$  dificultan aún más la ya compleja operación. Por tanto, la automatización de cualquier parte de este proceso puede constituir una importante mejora en el mantenimiento de este tipo de instalaciones, no sólo en cuanto a la reducción del tiempo de inspección y de los errores, sino también de los costes asociados.

Con este objetivo, en este artículo, se propone un sistema de control para realizar el guiado de un AUV (Autonomous Underwater Vehicle) capaz de detectar y seguir autónomamente un cable/tubería sobre la base de retroalimentación visual. Dicho sistema incorpora, adicionalmente, una novedosa estrategia de evitación de obstáculos/navegación que permite al vehículo escapar rápidamente de cualquier situación de aprisionamiento potencial.

El subsistema de visión, por un lado, aprovecha la característica rigidez y forma del cable/tubería para localizarlo en las imágenes que procesa. Múltiples secuencias de imágenes reales captadas con un ROV han sido utilizadas en su validación. Estas imágenes muestran cables parcialmente cubiertos por algas y arena estando a su vez rodeados por algas y rocas, lo que las hace altamente realistas. La tasa media de aciertos alcanzada ha sido de aproximadamente del 90% para una tasa de imágenes de más de 40 por segundo.

En cuanto a la arquitectura de control, siendo reactiva, está basada en comportamientos descritos como campos de potencial, todo ello con la intención de interactuar adecuadamente con un entorno dinámico y no estructurado como es el



submarino. Simplicidad, robustez y modularidad son las principales características de este enfoque. No obstante, de él también se derivan varios problemas entre los que destaca el de aprisionamiento o bloqueo debido a mínimos locales. Diferentes configuraciones de obstáculos pueden dirigir a esta situación, tal como el típico cañón en forma de U. Por esta razón, al sistema de control se le ha añadido en su etapa final un nuevo componente denominado *Random T<sup>2</sup>* que da solución al problema aplicando dos nuevos conceptos: *Traversabilidad* y *Tenacidad*. Básicamente, este componente, dependiendo de la configuración de obstáculos existente alrededor del vehículo, puede seleccionar como salida una dirección de movimiento diferente a la sugerida por las etapas previas del sistema. La estrategia ha sido comparada con otras aparecidas en la literatura que son habitualmente referenciadas: *Avoiding the Past*, *Learning Momentum* y *Micronavigation*. Para ello, se ha hecho uso de NEMO-CAT, un entorno de simulación 3D implementado con OpenGL que incorpora el modelo hidrodinámico de un vehículo real diseñado y construido por el grupo de investigación *Computer Vision and Robotics* de la Universidad de Gerona denominado GARBI.

## CONCLUSIONES

El presente artículo presenta una novedosa arquitectura de control para el seguimiento de cables/tuberías submarinas utilizando retroalimentación visual. Ésta incorpora una estrategia específica para evitar el típico problema de aprisionamiento característico de todos aquellos sistemas de control reactivos basados en el principio de campos de potencial. Una solución tanto a nivel local como global ha sido aportada haciendo uso de una reducida cantidad de recursos computacionales y de memoria. A su vez, un estudio comparativo de la eficiencia de la estrategia desde el punto de vista de la longitud del camino recorrido por el vehículo hasta su objetivo ha sido llevado a cabo. Como resultado, nuestro algoritmo generó, en media, trayectorias 2.4 veces más cortas.