AI-based Path Planner for an Autonomous Underwater Vehicle

GERARDO ACOSTA [1-2] - HUGO CURTI [3] - OSCAR CALVO [2] - SILVANO ROSSI [4]
[1] Investigador CONICET, ARGENTINA
[2] Grupo de Tecnología Electrónica – Dept. de Física – Universitat de les Illes Balears, SPAIN.
[3] INCA – Fac. de Cs. Exactas – Universidad Nacional del Centro de la Provincia de Buenos Aires, ARGENTINA.
[4] INTELYMEC – Fac. de Ingeniería – Universidad Nacional del Centro de la Provincia de Buenos Aires, ARGENTINA.

Abstract: - This article describes the design considerations and experimental results in the implementation of the strategy to generate trajectories for an autonomous vehicle. The problem approached consists on an autonomous underwater vehicle (AUV) tracking a pipeline in the seabed. To solve this problem, a real time expert system (named EN4AUV) was developed and included in the on-board AUV central processing unit. EN4AUV takes trajectory control decisions based on a number of variables, arranged around the concept of scenarios. For different scenarios, the expert system is able to suggest trajectories. Recent trials were performed successfully in the North Sea. The article presents the full system architecture comprising the dynamic mission planner and the navigation, guidance and control systems, paying special attention to the knowledge-based path planner (KBPP) within the mission planner. Results of its performance during these sea trials are discussed.

Key-Words: autonomous underwater robots, path planning, artificial intelligence, knowledge-based system, underwater pipeline inspection

1 Introduction

One of the most challenging applications of AUVs is the autonomous tracking of underwater pipelines, mainly for maintenance purposes. Even though nowadays these inspections are usually done with remote operated vehicles (ROV), this approach has two basic drawbacks: the poor quality of acquired data and the high cost incurred by the need of a mother ship, an extra positioning vessel, and their crew, each time an inspection has to be undertaken. The situation is even worse, as the surveys depths are increased, like in offshore petroleum exploitation. In contrast, AUV's allow a smoother navigation and then a more reliable data acquisition, since there is no umbilical cable to a ship or platform, and missions can be performed in a cheaper and faster way, since the infrastructure requirements are minimal. To achieve this, it is expected that such an AUV could reach positions in global coordinates navigating autonomously, with low position error. In addition, a mission planner should propose the trajectory to be followed by the vehicle's guidance and control systems [1] based on sensor readings.

Trajectory generation for mobile robots represents a hard problem due to the great amount of data required to make a decision in short time. This problem has attracted the AI community research for years. Three main lines of activity have appeared: (a) planning-based systems [2], [3], [4], (b) behavioral-based systems [5], and (c) hybrid systems [6].

2 Underwater pipeline inspections

The pipeline tracking with an AUV implies the following actions: the pipeline detection itself, and the use of this information by the vehicle's guidance and control systems to determine the desired trajectory to follow. These events should guide the AUV along the pipe at a predefined offset, without human intervention. If the AUV stops acquiring reliable good quality data, that is, its sensors acquires data below a predefined certainty, it is necessary to correct the trajectory or asses a new running hypothesis (i.e., the target is buried or there is an obstacle in the pipeline's trajectory).

In the present application, two types of target tracking were considered depending on the sensor utilized:

- Acoustic tracking, based on a multibeam echosounder (MBE)
- Magnetic tracking, based on a magnetic sensors system (MAG)

Both sensors, and a file containing historical data about the original pipeline location, act as an input to a sensor fusion module (SFM). This module yields an idea of the relative AUV/pipeline position.

In a real underwater mission, unforeseen situations like the sudden appearance of a fishing net, or a detour due to obstacle detection, may appear. Another objective of this R+D work was to provide an AUV with enough “intelligence” to cope with these real situations in the marine world. Then, a the expertise of expert ROV operators was elicited and codified in the form of a real time expert system.

3 The Knowledge-Based approach

Such expert system, named ***EN4AUV*** (Expert Navigator for Autonomous Underwater Vehicles), was built using CLIPS, a C based shell [7]. The classical steps followed were: (a) problem identification, (b) conceptualisation, (c) formalisation, (d) implementation, and (e) evaluation. As it is well known, these phases are progressive and there is a dynamic feedback among them during system development.

The problem to be solved by the Expert System is to generate the vehicle’s trajectory, based on the position coordinates (***x, y, z***) provided by a sensor’s module, and a confidence in the measurement of such co-ordinates, called **certainty error**. The EN4AUV then proposes a desired trajectory, formed by four points to be reached and surpassed by the submarine (waypoints). This desired trajectory is in global co-ordinates indicating latitude, longitude and altitude. Then EN4AUV is clearly a reactive expert system, behaving according to the situation, taking into account, for instance, the pipeline status, the type of survey, the different mission settings, and others. The concept of scenario was used to classify different situations.

3.1 Scenarios

A scenario is defined as a set of input variables that describe a situation. The AUV shall react in different ways from one scenario to another. Through data abstraction, the collection of scenarios may then be considered as the world model to solve situations. For the trials described in this article, fourteen scenarios were programmed. As a consequence of the scenario a set of few parameterised subtasks are fired: **findstart**, **search**, **backtostart**, **skip**, and **track**. A concatenation of these subtasks constructs the final AUV trajectory.

1st Scenario: The AUV is tracking an exposed pipeline, navigating on top, at a fixed offset smaller or equal than 5 meters. Both the MBE and the MAG can detect it.

2nd Scenario: The AUV is tracking a buried pipeline on top, at a fixed offset, smaller or equal than 5 meters. The MBE may not be able to detect it, but the MAG can track it anyway.

3rd Scenario: The AUV is tracking an intermittently exposed and buried pipeline at a fixed offset. This is a sequence of alternative appearance of scenarios number one and two. MBE, MAG and LD are used.

4th Scenario: The AUV is tracking a free-span pipeline at a fixed offset. The pipe is tracked mainly based on MBE readings, which may be detecting the pipe itself or the trench.

5th Scenario: The AUV is tracking a pipeline in the presence of one or more pipes (like infield pipelines) or other magnetic objects in the area. Measures from MBE as well as MAG are needed.

6th Scenario: The AUV is tracking a pipeline but avoiding an obstacle. In such scenario the certainty error may increase beyond its thresholds, but the EN4AUV knows where the pipe is and ignores the pipe_lost flag. The path planner module (PPM) outputs a flag indicating this condition and the EN4AUV may query the legacy data to confirm the existence of an exclusion zone. Although sensor readings are not reliable, they are not turned off to be ready when the AUV is again over the pipeline.

7th Scenario: The AUV is searching a buried pipeline. No readings from MBE, just MAG will yield detection when the AUV is right over the pipe. With two detection (crossing) points the pipeline direction vector is computed and the AUV starts tracking from the last known point with this direction.

8th Scenario: The AUV is searching the pipeline, which is considered as lost. EN4AUV shall have an estimate of the trajectory from SFM considering the whole inputs: MBE, MAG, SSS and LD.

9th Scenario: The AUV is searching a pipeline in the presence of one or more pipes (like infield pipelines) or other magnetic objects in the area. Every information source is operative to discriminate the target under study (MBE, MAG, SSS and LD).

10th Scenario: The AUV is skipping from one point to another. MBE, SSS and MAG are off to save energy. This special situation appears when changing from one pipe to another to track, or from one zone of interest to another over the same pipeline.

11th Scenario: The AUV is going back to the last known position to start tracking, after founding the pipeline as a consequence of a successful search. MBE, SSS and MAG are off.

12th Scenario: The maximum number of reacquisition after unsuccessful searches was reached. The mission is ended with a failure message.

13th Scenario: The AUV is tracking an exposed pipeline, navigating on top, at a fixed z_offset greater than 5 meters. The detection is done mainly with the

MBE.

14th Scenario: The AUV is tracking a buried pipeline on top, at a fixed z_offset greater than 5 meters. The blind tracking is done mainly based on legacy data, and cannot last more than half a minute. After this, if there are no more sensor readings, a new search must be started.

3.2 Rule base and objects

CLIPS allows the knowledge representation to be in the form of rules and frames (COOL or Clips Object Oriented Language). These formalisms are used in the knowledge base (KB) to represent the involved knowledge. Thus, there is a set of rules devoted to pipeline's layout determination (if it is or not detected, if buried or free span, etc.). Once this is assessed, rules determine the AUV "follow status" as regards as the pipeline. These "follow status" may be: avoiding an object, pipeline found, pipeline lost or pipeline intermittent. Then rules determine which scenario is present, and then select the corresponding action. These actions are modularly implemented as C++ routines. In Fig. 1, a rule for determining the AUV's follow status is presented in the CLIPS syntax and in Fig. 2, the class definition for the concept of "*working scenario*" is also shown.

```
(defrule R04.1 ; it seems we are lost, but we are
                          ;avoiding an obstacle
   (PPLS ready)
   ?ps <- (object (is-a SURVEY))
   ?ws <- (object (is-a WORKING_SCENARIO) (Avoiding ?av))
   (test (= 1 ?av))
=>
   (send ?ws put-Follow_status AVOIDING)
   (assert (R4SD))
   (printout t "CLIPSMACHINE: R04.1 Follow status is " (send ?ws
get-Follow_status) crlf))
```

Fig. 1. A knowledge-base rule from EN4AUV.

About 50 rules conform the current KBPP. They are forward chained as usual in data driven, real time expert systems. The inference rule used is based on the Rete's algorithm. The objects used for formalisation were: symbolic_variable, Waypoint, Type_of_Survey, Survey, Trajectory, Input_ Trajectory (is-a-Trajectory), Output_Trajectory (is-a-Trajectory), Object_of_Study (is-a-symbolic_variable set to pipeline in this first approach), Working_Scenario (is-a-symbolic_variable).

3.3 The runtime environment of EN4AUV

The KB system runs as an embedded application, interchanging messages with other modules, as it may be seen in figure 3, where the full system architecture is presented.

```
(defclass WORKING_SCENARIO (is-a SYM_VAR)
   (role concrete)
   (pattern-match reactive)
   (multislot Movie (type SYMBOL) (create-accessor read-write))
   (slot Navigation_type (type SYMBOL) (create-accessor read-write))
   (slot Incident_point (create-accessor read-write))
   (slot Error_Budget (type FLOAT) (create-accessor read-write))
   (slot Avoiding (type INTEGER) (create-accessor read-write))
   (slot Tracking_status (type INTEGER) (create-accessor read-write))
   (slot Follow_status (type SYMBOL) (create-accessor read-write))
   (slot Quantity_of_sensed_object (type SYMBOL)(create-accessor
read-write))
   (slot Risky_situation (type SYMBOL) (create-accessor read-write))
   (slot Within_corridor (type INTEGER) (create-accessor read-write))
   (slot Count_Reacq (type INTEGER) (create-accessor read-write))
   (slot Count_Rep_Reacq (type INTEGER) (create-accessor read-
write))
   (slot Error_distance (type INTEGER) (create-accessor read-write))
   (slot Navigation_Height (type FLOAT) (create-accessor read-write))
   (slot Search_results (type INTEGER) (create-accessor read-write)))
```

Fig. 2. A framework representing a Working Scenario.

Under GNU Linux, running in an industrial PC-104 computer, the remaining modules were developed in C++. Every module has an input-output data flow based on messages put on a UDP channel and broadcasted for the remaining ones, using TCP/IP protocol.

As it may be seen, EN4AUV receives position estimation from the sensors' module, can access to historical data and can receive information from a (low level in the control scheme) path planner. Signals from sensors are converted into an absolute depth, latitude and longitude within the SFM (***x, y, z***). This triplet is transmitted to the Path Planning module within a message. Once received, the content of this message is stored in a "Waypoint" object. According to the present scenario assessed by EN4AUV, and based on the waypoint received, it provides a desired trajectory formed by four waypoints. Then, the KBPP combines what is desired (the trajectory proposed by EN4AUV) with what is possible (trajectories from the obstacle detection system), completing the KBPP performance.

4 Experimental Results

The whole system was first tested in simulation, within a development environment such that the software could be directly ported to the vehicle when tests were successful [8]. This simulation environment was simple (without fancy interfaces) but realistic enough to validate most of the AUV´s software. This trade-off was made exercising the three basic working hypotheses that follows: (a) simple linear dynamic models of the vehicle's kinematics; (b) static information sharing among different modules, achieved through bottom specifications files (.bsf) and pipeline specification files (.psf); (c) dynamic information interchange among different modules, achieved with a communication framework based on messages.

These simulation tests were performed as part of the integration process of different modules in fig. 3. As modules were developed by different partners in different geographical locations, in the early stages, remote integration tests were performed using a tunneling system to send broadcast UDP messages from the local area network of each development partner to the others.

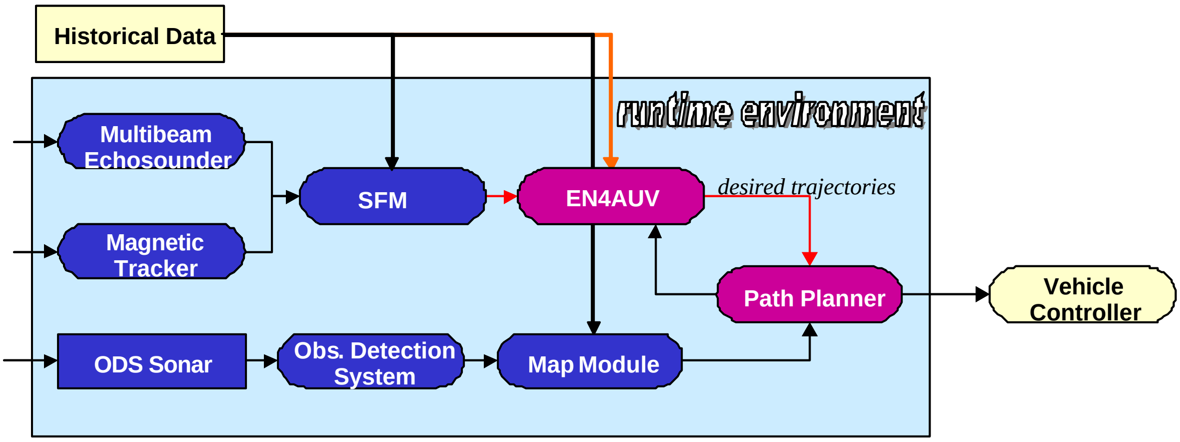

Fig. 3. Runtime environment for EN4AUV.

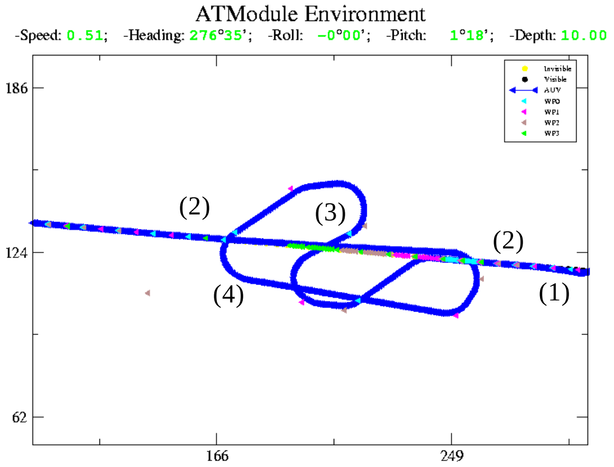

Fig. 4. Simulated mission 7 from Orkney trials.

For instance, the SFM-EN4AUV integration tests were performed to check their connectivity. This was a remote integration test where SFM ran on National Technical University of Athens's LAN (in Greece) while EN4AUV, ATMEnvs and the simulator graphic interface ran on UIB's LAN (in Spain). In Fig. 4 mission 7 performed in Orkney trials is depicted (see Table I). As it may be seen, a FindStart(1) – Track(2) – Search(3) – BackToStart(4) – Track(2) mission was executed.

After simulation tests like this, some sea trials were scheduled with the Subsea7's Geosub commercial AUV of Fig. 5. The first trial was performed at Peterhead (Scotland) from 18 to 23 August 2004. The second one was performed at Burray, Orkney Islands (Scotland) from 13 to 21 September 2004, and the results are presented in the following sections.

Fig. 5. The experimental AUV (Geosub)

4.1 Peterhead tests

The main purpose of these sea trials was to test the connectivity among modules developed by different partners. Particularly, EN4AUV's path planning strategies, SFM and the Geosub's native NGC systems were tested to evaluate their performance at sea. These tests were performed at surface in a bay. The simple MBETracer simulator, part of the simulation environment, was used to emulate a pipeline. Thirteen missions were run, and the main goals were completely achieved, even when each trial had different degree of success.

Unfortunately, there was a bug in EN4AUV initialization that caused a timing problem: the Adaptor could not make the replanning. Although the problem was tackled during trials, there was not enough remaining time to test it. Fig. 6 shows the results of mission 12 from the AUV's transmission received and registered in the dock by the Subsea7 tracker.

Fig. 6. Peterhead mission 12 depicted by S7 on-line screen tracker.

4.2 Orkney Islands tests

The objective of these tests was to verify the navigation system at deep water (mainly gyroscopes) and the tracking modules (multi-beam echo-sounder tracer and SFM) with a real pipeline, as well as to refine and test EN4AUV. The trials were performed in Scapa Flow, and the AUV operated over Talisman's 30" Piper/Claymore to Flotta pipeline.

EN4AUV was contrasted against a state-machine approach described in [9]. As it is summarized in TABLE I, in most of the cases there was a coincidence between the conclusions of both systems, performing as expected. This coincidence was because of the experiments done were not complex[1] enough to determine that EN4AUV has a superior performance than a simple and robust state-machine. In addition, there was not enough time to perform a full deep-water test, so the original objectives were partially achieved. After these final trials in the late 2004, the need of more trial sessions before putting the whole AUTOTRACKER-Geosub into production state turned to be obvious. In effect, 3D EN4AUV mission planning was completely developed and tested on the simulator, but could not be tested at sea yet. However, the successful results showed us that the technological research is going in the right direction.

In Fig. 7, the AUV trajectory during Mission 15, as seen from the screen of the SeeTrack[2].

TABLE I
Orkney Missions Summary

MISSION	GOAL	STATUS	DESCRIPTION
1 to 5	Native AUV NGC systems and hardware tests	Partial Success	Problems in AUV internal network arose and the information from the sensors could not be acquired.
6	Track-Search-Back2start-Simulated pipeline. Consistency between EN4AUV and SM	Failed	A configuration error in SFM caused EN4AUV to abort the mission.
7	Track-Search-Back2start-Track with EN4AUV. Simulated pipeline	Success	EN4AUV completed the mission. The trajectory that the AUV performed was almost the same as the one predicted by the simulator ATMEnvs.
8 to 13	Simple Track of a real pipeline	Failed	Many parameters adjusted between SFM and MBETracer to perform a correct pipeline detection.
14	Simple Track of a real pipeline	Aborted	A configuration error caused this mission to abort before starting.
15	Simple Track of a real pipeline. Consistency between EN4AUV and SM	Success	EN4AUV completed the mission. The pipeline was correctly tracked.

5 Conclusion

Some important hints about the fundamental considerations in designing the mission replanning system, in particular, the path planner for an existing AUV, the Geosub from Subsea7, were presented in this article. One of the most outstanding features of EN4AUV is that its knowledge-base may be enlarged as new experiences from more difficult cases are elicited. The addition of new cases in the KBPP's KB implies the addition of new rules. Then the attention must be focused on the KB consistency, as with any expert system. Also, this system design allowed developing modular KB (i.e., one KB for pipeline

---

[1] Only the MBE was fully operative for trials. SSS and MAG were not available. The pipeline was mainly straight direction and exposed.

[2] Courtesy of SeeByte LTD.

status determination, one for scenario determination, one for action calling), with a great debugging facility during implementation. Software's reusability is also enhanced by this approach. Effectively, the same system may be used to submarine cable tracking, to coastal studies, and other applications simply interchanging different KBs.

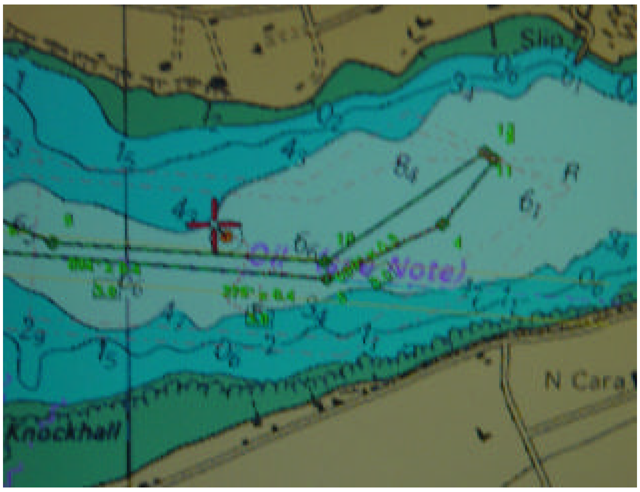

Fig. 7. Mission 15, as seen from the screen of the SeeTrack

Using on-line acoustic signal processing showed to be very advantageous as regards as other approaches for pattern recognition in the underwater world. Compared to visual images, it is less energy consuming and does not depend on water transparency. For deep-water inspections, undoubtedly sonars are the appropriate sensors to be considered in the payload.

Although there are pending tests, the sea trials results reported here, settle a sound basis for a ready-to-transfer prototype. They demonstrated that the technology is mature enough to face the challenge of autonomous underwater inspections with minimum human intervention in deep waters, as it was the goal of the project presented in this article, from the beginning.

6 Acknowledgements

This research was supported by the AUTOTRACKER Project within the 5th European Community Framework Programme (Project GDR1-2000-25150), and the British Petroleum Co. during the sea trials at Peterhead and Orkney.

This research was also partially supported by a Marie-Curie International Fellowship within the 6th European Community Framework Programme (Project 03027 – AUVI).

References:

[1] Fossen, T.: "*Marine Control Systems*", Marine Cybernetics AS, Trondheim, Norway, 2002.

[2] Chatila, R. and Laumond, J.P.: "*Position referencing and consistent world modeling form mobile robots*", Proc. of the IEEE Int. Conf. Robotics and Automation, St. Louis, MO, March 1985, pp 138-145.

[3] Albus, J.: "*Outline for a theory of Intelligence*", IEEE Transactions on Systems, Man and Cybernetics, vol. 21, Issue 3, May-June 1991, pp:473 - 509

[4] Albus, J.S.: "*The role of world modeling and value judgement in perception*", Proc. of the 5th IEEE International Symposium on Intelligent Control, Vol. 1, 5-7 Sep 1990, pp. 154-163.

[5] Brooks, R.: "*A Robust Layered Control System for a Mobile Robot*", IEEE Journal of Robotics and Automation, Vol. RA-2, Nº 1, March 1986, pp. 14-23.

[6] Payton, D., Kenneth Rosenblatt, J., and KEIRSEY, D.: "*Plan Guided Reaction*", IEEE Trans. on System, Man and Cybernetics, Vol. 20, Nº 6, November/December 1990, pp. 1370-1382.

[7] Giarratano, J.: "*CLIPS User's Guide*", NASA LB Johnson Space Center – Software Technology Branch, 2002.

[8] H. Curti, G. G. Acosta, O. Calvo: "*Autonomous Underwater Pipeline Inspection in AUTOTRACKER PROJECT: the Simulation Module*", IEEE/Oceans'05 Europe Conference, Brest, France, June 21-23, 2005, pp. 384-389.

[9] G. G. Acosta, H. J. Curti, and O. A. Calvo, J. Evans, D. M. Lane: "*Path Planning in an autonomous vehicle for deep-water pipeline inspection*", IEEE Transactions on Ocean Engineering (submitted for publication).