*Journal of Maritime Research, Vol. VI. No. II, pp. 37-52, 2009*
*Copyright © 2009. SEECMAR*

*Printed in Santander (Spain). All rights reserved*
*ISSN: 1697-4840*

# LOW-COST AUTONOMOUS UNDERWATER VEHICLE FOR UNDERWATER ACOUSTIC INSPECTIONS

O. Calvo[1*], A. Sousa[2*], J. Bibiloni[3*], H. Curti[4*], G. Acosta[5] and A. Rozenfeld[6]

Received 23 September 2008; received in revised form 30 September 2008; accepted 5 May 2009

## Abstract

**The design considerations of a low-cost Autonomous Underwater Vehicle (AUV) are presented in this article. Periodic surveys are needed in the preventive maintenance of submarine structures of offshore industries. The advantages of performing them with AUVs instead of Remote Operated Vehicles (ROV) or Towed Unmanned Devices are the lower costs involved and a better data quality in the inspection missions. Through EU funded projects (AUVI and Autotracker), and Spanish funded project IOGECS, the construction of a low-cost prototype for depths of 100m AUV was initiated. Previous results and design considerations and proposed solutions, as well as a description of the hardware employed, the sensors in the payload, and the mission replanning, are described in this work.**

**Keywords**: Autonomous Underwater Vehicles (AUV). Trajectory planning. Knowledge-based systems. Underwater pipeline inspection.

## INTRODUCTION

One of the most outstanding applications of autonomous underwater vehicles (AUV) is the inspection of submerged structures for maintenance purposes. These

[1] Professor. University of the Balearic Islands (oscar.calvo@Uib.es), Physics Department, Palma de Mallorca, Spain. [2] PhD Student, University of the Balearic Islands (andre.sousa@uib.es), Physics Department, Palma de Mallorca, Spain. [3] Graduate student, University of the Balearic Islands (jaume.corro@gmail.com), Palma de Mallorca Spain. [4] PhD Student. University of the Balearic Islands (hcurti@fio.unicen.edu.ar), Palma de Mallorca, Spain. [*] GTE Group, Physics Dept., Universitat de les Illes Balears-UIB, Spain. [5] Professor. Universidad Nacional del Centro (gerardo.acosta@ieee.org), CONICET and INTELYMEC Group, Electromechanical Dept., Engineering Faculty, Universidad Nac. del Centro Prov. de Buenos Aires-UNCPBA, Argentina. [6] Researcher. Universiti o the Balearic Islands IFISC (alex@ifisc.uib.es), Palma de Mallorca, IMEDEA-UIB, Spain.

inspections, performed to avoid infrastructure damage and to preserve the ecosystem, are usually done with remotely operated vehicles. This approach has two basic drawbacks when compared to untethered vehicles: the low quality of acquired data due to the umbilical perturbation worsens with depth, and the operational costs of mother ship and its crew.

**Figure 1a**. Geosub AUV ready for trials.

During the last few years great advances had been made in inspections using AUVs. As an example, the Twin-burger 2 [Balasuriya et al, 1998], guided by camera images presented some interesting results. But, offshore oil exploration is moving into deep waters, since oil reservoirs near the cost are starting to dry. In deep and opaque waters, video becomes less practical for AUV inspections due to the lightening requirements; power needs and light scattering effects. In these cases it is preferable to use sonar or a fusion of many sensors as described in [Tena Ruiz et al, 2006].

The first commercial AUV capable of locating and following a pipeline with acoustic sensor was developed in the framework of an EU funded project: AUTOTRACKER (GRD1-2000-25150) [Acosta et al, 2005]; [Curti et al, 2005], in which the authors participated, along with oil and cable companies like British Petroleum (BP), Alcatel, Seas Power and Subsea7, and with a team of universities such as then UIB, Heriot Watt University (HWU) and the National University of Athenes, (NTUA). The project consisted on modifying a commercial AUV (Geosub by Subsea7) (figure 1a) with the purpose of developing autonomous searching, locating and tracking of submerged oil pipes, partially buried or exposed, using acoustic and magnetic sensors. The GTE at UIB developed an Expert System that was responsible for the real time path planning and re-planning based on the information of the sensor. We also built a HIL simulator that assisted in the development of the rules and to try all possible situations. The vehicle utilized was the Geosub AUV (Figure 1), designed to operate in water depths to 3000 m and with mission times of between 30 and 60 hours, depending upon payload configuration. The vehicle measures 6.82 m long and 900mm in diameter, and weights 2400 kg. This vehicle is an ideal stable platform, operating with a comprehensive range of survey sensors close to the seabed. Experimental trials were performed at the North Sea in Scotland (Orkney

Islands) in which the UIB participated. A trajectory of a search and locate of the real pipe is shown in figure 1b. Companies are now exploiting the technology developed. Driven by these successes, we applied to the EU and the Spanish government for further funding and they were both awarded as a Marie Courie Fellowship (6PM IIF2006 Ref. 3027) and by the former Spanish Ministry of Science and Education under the Transport National Program (TRA2006-13318), in collaboration with the Ocean System Lab at HWU. We decided then to build a cheaper prototype, the AUVI vehicle, which would navigate mainly in shallow waters but allowing us to test computational intelligence algorithms for planning and replanning of vehicle's trajectories and tasks.

## AUVI PROTOTYPE DESIGNED FOR PIPELINE SEARCH AND INSPECTION

### Working hypotheses

The central working hypothesis was to design a low cost and easy to deploy AUV, that could be used as a testbed for different control algorithms and pipeline detection routines. It should be software compatible with the previous platform used in the Autotracker project, where most of the pipe tracker was developed, and it should run under the OceanShell environment developed jointly with the researchers from Heriot Watt University [Acosta et al, 2005] [Evans, 2003]. The same software that runs on the vehicle can run on a real time simulator previously developed by UIB [Curti et al, 2005] (Hardware in the Loop). Even though the Geosub AUV is an ideal platform for commercial autonomous inspections, being capable of diving at great depths with great accuracy and fairly good autonomy the operational costs of the trials were too expensive for a University budget. Then, the idea of having a shallow water AUV to test all the algorithms was very appealing, to minimize the costs of each sea trial. A light AUV capable of carrying the necessary sensors for specific missions, with a plug and play feature would be ideal to validate the algorithms product of our research. It should be backwards compatible with the software that runs on the full depth AUV.

### Mechanical description

The AUV was built using two torpedoes with a diameter of 225 mm. Each of the cylinders has at one end a half sphere with 225 mm of radius, it's the surface attack, and the opposite end is shaped like a cone with 225 m on your biggest diameter and 85 mm on your smallest one. The two bodies matching the structure of the vehicle are linked with metal tubes keeping at a distance of 430 mm between the centers of the cylinder.

The current design of the vehicle was based as a strategy to increase the passive stability and to optimize the torque on the thrusters, resulting in better navigation control given the interaction of the vehicle with the environment

| Prototype AUVI GTE-UIB | |
|---|---|
| Length | 78.74 cm |
| Width | 81 cm |
| Max depth | 52 meters |
| Weight without batteries | 16.5 kg |
| Weigth with batteries | 39.5 kg |
| Recharging time | 4 to 6 horas |
| Batteries | Sealed lead GEL |
| Capacity | 6 x 12v 33 AH |
| Autonomy | 75 to 150 minutes |
| Speed | 1 a 2 m/s |
| Motor thrust | 2x13.64 kg (horiz) |
| Motor Type | 4 12V DC motors |
| Comunicaciones | Radiomodem 115 Kbps/WiFi |
| Positioning | GP, dead reckoning, compass and Accelerometer |
| Manual Control | Joystick via UHF |
| CPU sensores | dsPic |
| CPU control Naveg. | Linux PC104 |
| Distance to bottom | AUTOHELP ST30 (NMEA) |
| Extra payload<br>Payload sonar<br>Depth sensor | 15 kg<br>Tritech SeaKing<br>Duck PTX 1400 |

**Tabla I.** Main Specs of the AUVI.

## Hardware Architecture

Two main processor boards compose the AUV hardware: a low level and a high level electronics. The low level electronics is based on an Ingenia microcontroller board: the iCm4011 board, that uses a dsPIC 30F4011 as main processor. This board provides $I^2$ C and RS232 interfaces to communicate with sensors and high level CPU. The high level board is an intel X86 single board computer running LINUX made by Compulab, a 5Watts small footprint PC. This is where most of the planning and controlling software resides.

The microcontroller board gets the position either from the GPS when surfaced, or from an accelerometer, integrating both x and y readings. The GPS sends its data using RS232 streams following NMEA format. The accelerometer is based on the Analog Devices ADXL 202, dual axis accelerometer. This inexpensive device measures accelerations with a full-scale range of ±g. The outputs are analog voltages or digital signals whose duty cycles (ratio of pulse width to period) are proportional to the acceleration. The duty cycle outputs can be directly measured by the microprocessor. The MBE sonar (Tritech Seaking) converts it readings into RS232 streams following proprietary data format. The Obstacle avoiding sonar, an Autohelm ST30 depth finder send measures acoustic echoes to the seabed and send it to the microprocessor using Seatalk protocol, (NMEA) by RS422 electrical signals that can be easily turn into an RS232 signal. The microcontroller communicates with the Linux host by USB. The digital compass, a CMPS03 robotics board, based on the Philips KMZ51 magnetic field sensor, provides the AUV orientation with respect to earth magnetic field. This device provides a PWM signal with the positive width of the pulse representing the angle. It interfaces directly with the dsPIC. The vertical propellers are attached to DC motors controlled by an MD22 - 50V 5A Dual H-

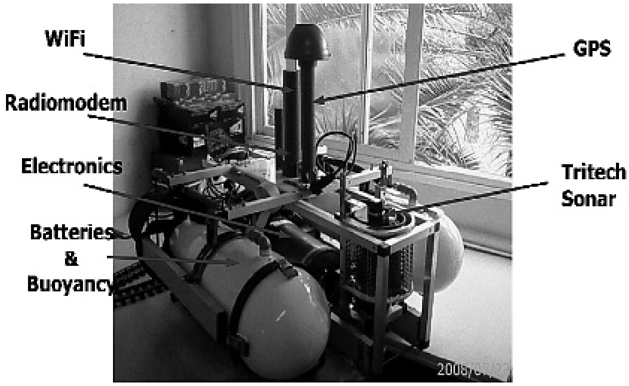


**Figure 2.** Experimental prototype.

Bridge made by Robot Electronics. The MD22 controller accepts I2C among the possible control signals. The two horizontal thrusters are driven by MD03 motor controllers, that were modified to accept direct control of the PWM signals from the dsPIC. All signals are opto-coupled for noise immunity. Isolation amplifiers are also used to measure motor speed and battery charge. An alarm circuit prevents water leaks. The electronics resides in a steel high pressure hull. Batteries are stored inside the torpedoes. A picture of the prototype is shown in figure 2 and a block diagram of the electronics can be observed in figure 3.

**Software development and runtime environments**

The central idea of this development was to employ the same environment for software development than for runtime. Also, special emphasis was placed in keeping backward/forward compatibility between the commercial AUV with the shallow water prototype. A modular philosophy was used in the design and development of the small AUV, re-using many modules previously developed and tested in other applications. This software architecture, running on a Debian Linux CPU, is based on communications messages between modules implemented using the UDP layer. The system is divided in modules specialized in a particular task. The software is based on messages going from sensors to actuators. The pipe is detected with acoustic, optical or magnetic sensors. Positioning data obtained by GPS, INS or DVL are fused in the sensor fusion module to provide accurate position of the pipe and vehicle. This information is used by the Dynamic Mission Planner that decides the trajectory to follow. That trajectory is verified against the exclusion zones and probable obstacles detected by the Obstacle Avoidance Sonar (OAS) and modified accordingly before is sent to the path planner. The path planner also receives information from a Static Mission planner, that could take control at any time modifying the final trajectory (i.e. safety reasons, beginning of mission, abort, etc). Finally the actual command is sent to the Autopilot that controls thrusters, rudders and buoyancy.

The simulation environment could communicate with the vehicle through the serial port, using an adaptor between the NMEA protocol and the proprietary communication protocol among modules. In this way, once the software is operative, the

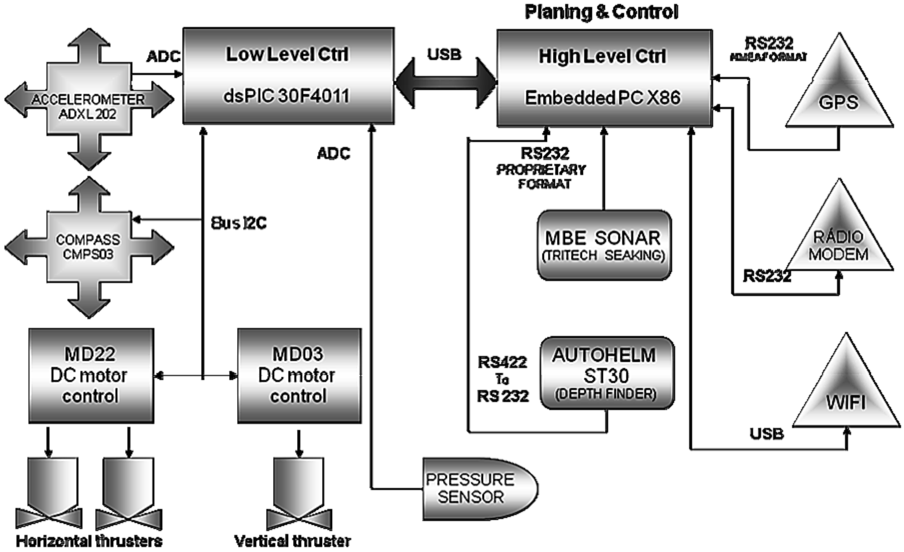


**Figure 3.** AUVI Electronics.

simulator is easily and straightforward replaced by the real hardware prototype. The adapted architecture for this prototype is shown in Figure 4.

The dynamic Mission Planner (DMP) [Acosta et al, 2005] and the Obstacle Avoidance Software (OAS) are in cascade. Thus, if the OAS does not detect any object through the forward-looking sonar, its output will be simply the desired trajectory from the DMP. On the contrary, if an obstacle is detected, the OAS changes the necessary waypoints in the trajectory provided by the DMP.

The tracking system employs a TriTech Seaking imaging sonar. For obstacle avoidance we replaced the multi-beam sonar by a low cost ST-30 sonar. In the first dry trials a Garmin GPS and an integrated compass were used to acquire the 2D position (x,y) and the yaw y, respectively.

As regards as the observer module in Figure 4, it simply listens to the position messages broadcasts and shows the vehicle's trajectory on the screen. The human machine interface is also employed to set the mission settings and start the simulation. A Graphical User Interface was built and it is shown in figure 5. UDP messages between threads are shown in the interface. This is a useful feature for debugging or to assist the operator when the AUV navigates on surface and radio link is available. The general architecture of an AUV used for pipe detection works with the dataflow going from sensors to actuators. The pipe is detected with acoustic, optical or and magnetic sensors. Positioning data obtained by GPS, INS or DVL are fused in the sensor fusion module to provide accurate position of the pipe and vehicle. This information is used by the Dynamic Mission Planner that decides the trajectory to

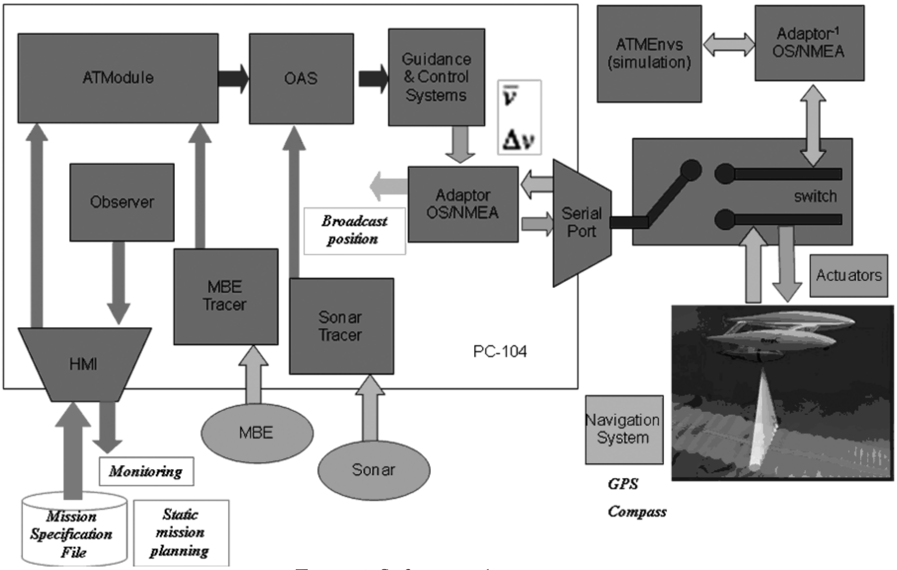


Figure 4. Software architecture.

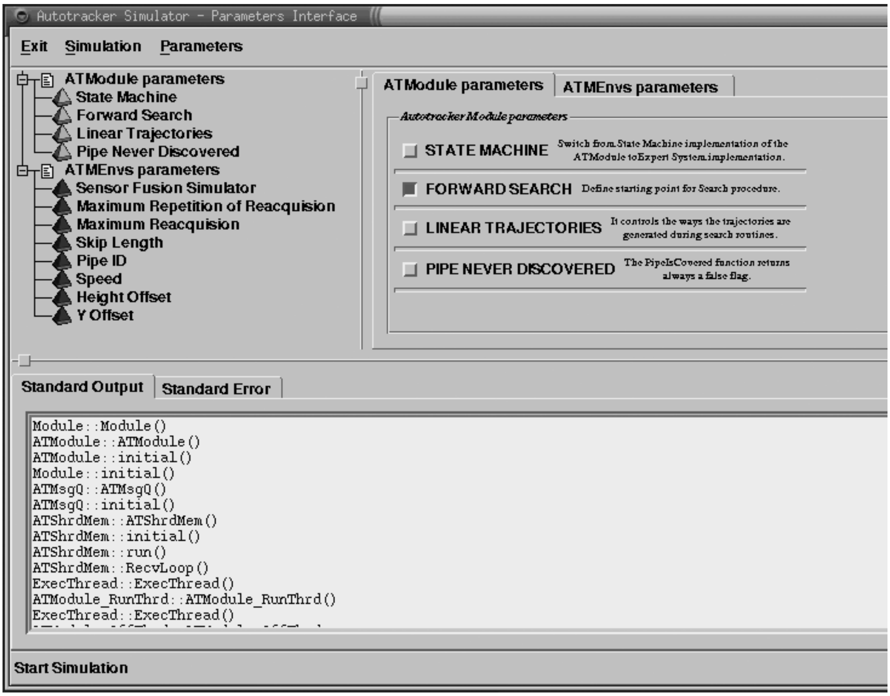


Figure 5. Graphical User Interface for mission planning.

follow. That trajectory is verified against the exclusion zones and probable obstacles detected by the Obstacle Avoidance Sonar (OAS) and modified accordingly before is sent to the path planner.

The path planner also receives information from a Static Mission planner that could take control at any time modifying the final trajectory (i.e. safety reasons, beginning of mission, abort, etc). Finally the actual command is sent to the Autopilot that controls thrusters, rudders and buoyancy.

## Dynamic path Planning and re-planning

The dynamic path planner (DMP) must provide the desired vehicle trajectory formed by a series of waypoints. New waypoints are decided based on pipe estimates coming from the Sensors and different situations. The DMP software contains built-in reasoning to mimic human surveyors decisions. Its switches from different states: find/track/skip/reacquire the target or abort the mission. The software controls the AUV to descend, acquire the pipe/cable, follow it at a given offset, speed, .etc, reacquire. Two different mission planners were built: a state based and an Expert system based. The State machine approach is basic and simple what makes it very robust. On the other hand the Expert System based is more flexible, expandable and incorporates the ROV operator's knowledge in the form of rules.

## State Machine path planning

The state machine implementation can be seen in figure 6. The vehicle is dropped in the water and navigates towards known coordinates where search operation will begin. The AUV starts the search navigating in zig-zag within a corridor known from legacy data. If the pipe is found it will track it with its sensor. If the pipe is lost it will search it again in the same area or in a different area after skipping a zone. This process can be repeated a programmable number of iterations, defined in the mission planning.

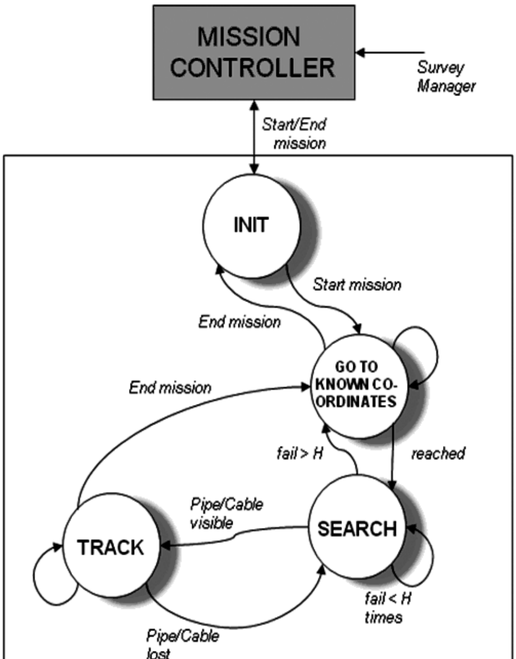


**Figure 6.** State Machine approach.

## Expert System based dynamic mission planner (DMP)

When performing an inspection, many situations might come up (like the sudden appearance of a fishing net, a complex pattern of more than one pipeline over the seabed, or a detour due to obstacle detection, and many others) in which the DMP should exhibit an "intelligent" behavior. To cope with these real situations in the marine world, we resorted to the experience and skills of ROV operators. A little part of their knowledge was elicited and codified in the form of a real time expert system. This expert system, named EN4AUV (Expert Navigator for Autonomous Underwater Vehicles) [Acosta et al, 2006], was constructed using CLIPS, a C based shell. Its main feature is to assess a current situation to act accordingly, in a clearly data driven/reactive behavior. Thus, EN4AUV is a reactive expert system, taking the proper action for every different situation, and considering the pipeline/cable status, the type of survey, the different mission settings, and others. These situations were coded as possible scenarios, in about fifty rules, like the one presented in Figure 4. As the knowledge about different situations increases, the knowledge base (KB) describing new scenarios can be completed and updated, yielding an incremental KB growth. Each scenario triggers different searching or tracking strategies, which are then subtasks with their own features. Scenarios are based mainly on two ideas: the survey types and the tracking states. The type of survey is defined a priori in the SMP, to establish the number of pipelines/cables to be tracked, the navigation depth, and other mission features. The other basic component of the scenario determination is the tracking state that changes when the SFM updates its sensors. From this, the EN4AUV may decide the status of the pipeline/cable (if buried, exposed, intermittent or freespan) and how is the AUV as regards as the pipeline/cable (if avoiding an obstacle, with the object under study considered as found or lost, or returning to a previous known position). Once scenarios are established, a typical situation assessment task, EN4AUV must output a desired trajectory or must decide a pipe/cable reacquisition. To yield a desired trajectory, the actions are organized in a set of few simple subtasks: findstart, search, back to start, skip, and track. Then the final trajectory of the AUV is built by one of these subtasks, or by a concatenation of them.

## The Knowledge-Base

Such expert system, named EN4AUV (Expert Navigator for Autonomous Underwater Vehicles), was built using CLIPS, a C based shell [Giarratano, 2002]. The classical steps followed were: (a) problem identification, (b) conceptualization, (c) formalization, (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. Examples of scenarios developed for the sea trials were the following:

### 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 parameterized 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.

```
(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)))
```

**Figure 7.** A framework representing a Working Scenario

```
(defrule R05.1
      (R4SD)
      ?a <- (R4SD)
      ?ws <- (object (is-a WORKING_SCENARIO)
(Count_Reacq ?cr)(Search_results ?sr) (Movie $?MOVIE)
(Follow_status ?FS))
      ?sv <- (object (is-a SURVEY) (Max_Reacquire ?mx))
      (test (<= ?cr ?mx))
      (test (eq ?FS LOST))
      (test (eq (send [OBJ_STUDY] get-
Present_Layout_Status) NOT_DETECTED_NOT_BURIED))
      test (= 0 ?sr))
=>
      (assert (Current_scenario SC8))
      (assert (PPLS notready))
      (retract ?ws)
      (insert$ $?MOVIE 1 SC8)
      (send ?ws put-Movie $?MOVIE)
      (retract ?a) ;to avoid the assertion of multiple scenarios in
the same KB query
      (printout t "CLIPSMACHINE: R05.1 Current Scenario is
SC8 -searching a pipe/trunkline" crlf)
      (printout t "CLIPSMACHINE: inserting movie" $?MOVIE
crlf))
```

**Figure 8.** A rule from the knowledge-base of the DMP for the 8th scenario determination, in the typical CLIPS syntax.

## 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. 7, the class definition for the concept of "*working scenario*" is also shown and a rule for determining the AUV's follow status is presented in the CLIPS syntax in Fig. 8.

About 50 rules form 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 formalization 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).

## EXPERIMENTAL RESULTS

The performance of the AUV navigation was tested with two different methods. The first one is a simple control method based on the line of Sight (LOS). With this simple scheme the AUV was able to follow a series of waypoints minimizing the angle computed as the difference between the actual heading and the angle of the line that joins the AUV with the target (next waypoint). A corrective signal proportional to the heading error is applied to the thrusters as a differential speed, adjusting the heading continuously to minimize this heading error. The scheme of the method is presented in figure 9. The control system in its present state is a simple proportional-integral (PI) control in the horizontal plane (2D) minimizing the angle β.

Recently, this DMP was integrated in the AUV described in this article. Preliminary guidance and control system approaches were tested in the new prototype. These trials were performed in shallow water, with the vehicle navigating just below water,

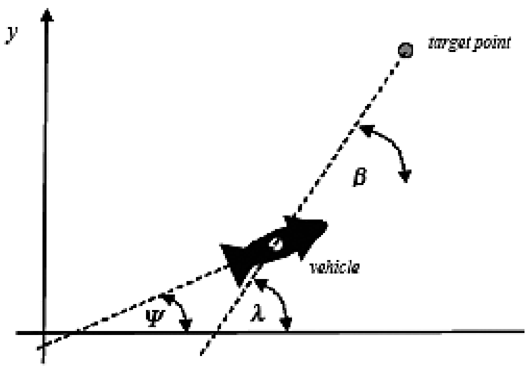

**Figure 9.** LOS navigation method.

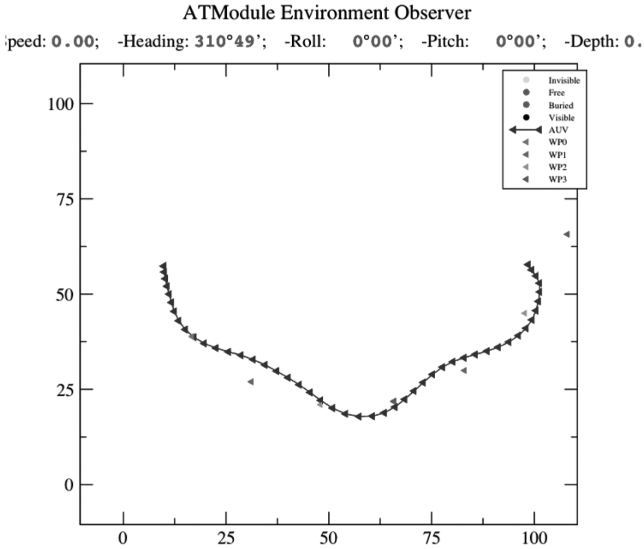


**Figure 10.** Sea trial of the AUV prototype in shallow Sea trial of the AUV prototype in shallow waters in Mallorca, Spain. Circular path.

obtaining position by GPS. In the first test a series of waypoints over a curved path was sent as a reference to follow using the line of sight algorithm (LOS). The desired path is a smooth trajectory that goes over the waypoints shown in figure 10, while the continuous blue line shows the real AUV positions. The area where the test was performed was quite limited and the controller was a simple non-optimized proportional producing some detours from the desired path. The AUV was programmed to go through a series of eight waypoints inscribed in a circle with a radius of around 45m. The vehicle followed the waypoints correctly though some oscillations were observed when passing over waypoints 2 and 4. It should be noted that an error tolerance was given to the control since the experimental area was small and turning radius was limited by software to avoid oscillations. Since the aims of the tests were to determine the maneuvering of the vehicle in water a proportional controller was used at this stage. Better results should be expected with a PI or PID controller. Trials were made in a shallow bay called Cala Estancia, near Palma de Mallorca, at the Balearic Islands. A second trial of the SEARCH condition was performed later. The search of a buried pipeline (emulated) was performed, as shown in figure 9. The sonar was simulated since the exact position of the pipe was known but it was assumed that the sonar would not see it. The vehicle starts at (58 m, 0 m) and moves to the initial position to start the search process at about (65 m, 25 m). This time the vehicle has more inertia than in the previous trial due to extra paylod electronics, this produces a couple of extra loops to reach to the start of search position. The pipeline is shown as a yellow continuous line in this figure. Since the pipeline is invisible the DMP decides to perform a search within a corridor. A series of waypoints describing a zig-zag movement are sent by the DMP as desired trajectories for the AUV's control system. These waypoints were joined by red straight lines in figure 11 for clarity. The AUV follows them reasonably well, though an improvement to the controller and a large area of tests should be done to give better results.

These experiments were performed on April 2008. A more sophisticated control system was designed combining a Lyapunov controller with the virtual particle path following strategy. Promising results were obtained through simulations and experimental test will follow soon. The control is composed by two blocks that allows to reach and track a pre-specified trajectory in real time, even in the presence of perturbations. The Control scheme is constituted by a Lyapunov based navigation block and a PI controller. The former creates robust references in order to guide the vehicle towards the desired trajectory.

## CONCLUSIONS

This paper describes in detail hardware and software aspects of an inexpensive AUV developed for pipe/cable tracking for inspection.

The key aspects of the design consideration of the software and hardware elements in the AUVI prototype were presented in this article, as a low-cost adaptation of the AUTOTRACKER vehicle. In particular, the dynamic mission planner based on an expert system was described. This mission planner showed that this artificial intelligence approach is able to re-plan the vehicle trajectory while in the mission, taking into account the mission settings, the changing underwater environment and the situation of the target under inspection. The new low-cost experimental prototype presented here, will surely be an adequate test-bed for the new task and path planning algorithms. In particular we will focus in a near future in the enlargement of the knowledge-base within EN4AUV for untested use-cases. Design details of a dynamic path planning with state machine and AI based technique were also shown.

Once again, rule-based approaches show that incremental prototyping is an easy way to code user's experience. Besides, coding knowledge, easy to understand for an experienced ROV user is fairly simple. The design is reusable since it can be applied to different targets (pipelines, power cables, telephony cables). This inexpensive AUV was developed to test the algorithms that allow

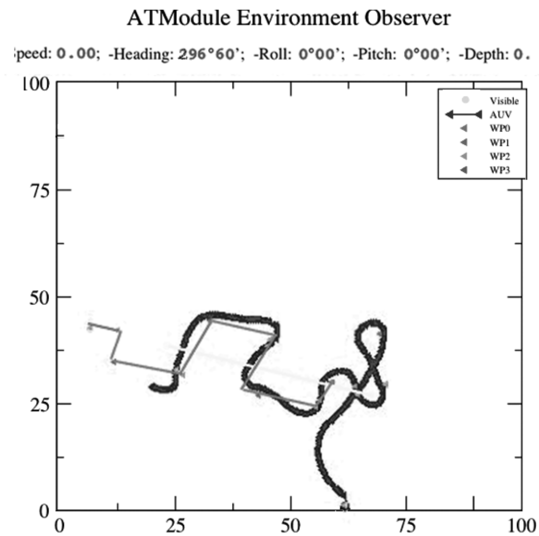


**Figure 11.** Sea trials of the AUV prototype in shallow waters in Mallorca, Spain. Searching a pipe.

the detection and tracking of the pipe with acoustic sensor. The paper presents a control method to govern the AUV performing lawmower searches of pipelines and cables. The simple method, is based on the LOS algorithm and was tested in a prototype AUV, passing through a series of waypoints. A more sophisticated control, based on Lyapunov to achieve better performance with guaranteed convergence was tested in simulations. Work is in progress to mount the MBE and Sidescan sonar and test the image recognition algorithms. The new designed vehicle will perform full 3D navigation so the control algorithms well have to be redesigned accordingly.

## ACKNOWLEDGMENTS

The authors acknowledge financial support by the MEC (Spain) and FEDER (EU) through projects TRA2006-13318 and PCI2005-A7-0356, the University of the Balearic Islands, the University of La Plata, CONICET and ANPCyT.

## REFERENCES

Acosta, G. G. , Curti, H. , Calvo, O. (2005): Autonomous Underwater Pipeline Inspection in AUTOTRACKER PROJECT: the Navigation Module, *Proceedings of IEEE/Oceans'05 Europe Conference,* June 21-23, Brest, France, pp. 389-394, Vol. 1.

Acosta, G. G., Curti, H., and Calvo, O. (2006): A Knowledge-based approach for an AUV Path Planner Development, *WSEAS Trans. on Systems,* Issue 6, Volume 5, pp. 1417 – 1424

Balasuriya and T. Ura (1998): Autonomous Target Tracking by Underwater Robots Based on Vision, *IEEE Proc. Int'l Symposium on Underwater Technology*, 15-17 April, Tokyo, Japan, pp. 191-197.

Curti, H. J., Acosta, G. G., Calvo, O. A. (2005): Autonomous Underwater Pipeline Inspection in AUTOTRACKER PROJECT: the Simulation Module", *Proceedings of IEEE/Oceans'05 Europe Conference,* June 21-23, Brest, France, pp. 384 - 388, Vol. 1.

Evans, J., Petillot, Y., Redmond, P., Wilson, M. and Lane, D. (2003): AUTOTRACKER: AUV Embedded Control Architecture for Autonomous Pipeline and Cable Tracking", *Proceedings of IEEE-OCEANS*, Sept., Volume 5, 22-26, pp. 2651-2658.

Fernández León, J. A.; Acosta, G. G.; Mayosky, M. A., and Calvo Ibáñez, O. (2008): A Biologically Inspired Control based on Behavioural Coordination in Evolutionary Robotics", Book chapter 7 in *Advancing Artificial Intelligence through Biological Process Applications,* Idea Group Inc.

Fossen, T. (2002): Marine Control Systems, *Marine Cybernetics AS,* Trondheim, Norway.

Giarratano, J. (2002): *CLIPS User's Guide*, NASA LB Johnson Space Center – Software Technology Branch.

Tena Ruiz, Y. Petillot, D.M. Lane (2003): Improved AUV navigation using side-scan sonar, *Proceedings of IEEE OCEANS,* Volume: 3 , 22-26 Sept. pp. 1261 – 1268.

Valenciaga, F., Puleston, P. F., Calvo, O., Acosta, G. (2007): Trajectory Tracking of the 'Cormoran' AUV Based on a PI-MIMO Approach, *IEEE/Oceans'07 Europe Conference*, Aberdeen, UK, June 18-22.