Tracking Benthic Boundaries Using a Profiler Sonar: a mixture model approach

Christian Barat **Maria João Rendas**
Laboratoire I3S (Informatique, Signaux et Systèmes de Sophia Antipolis)
2000 route des Lucioles, BP 121, 06903 Sophia Antipolis cedex, FRANCE
barat@i3s.unice.fr rendas@i3s.unice.fr

Abstract: **The paper presents signal processing and control algorithms that enable autonomous tracking of boundaries between distinct benthic regions by an AUV equipped of a profiler sonar. A novel sonar classification algorithm is presented, which uses the signature of the ocean floor in the incoming profiles to discriminate between distinct sea-bottom materials. By exploiting sonar scans of the region below the robot, a classical control loop is closed around the sonar data, using a feedback signal that is robust with respect to classification "noise".**

I. INTRODUCTION

The ability to track natural boundaries defined in the ocean floor by distinct habitats is useful in several applications, either military (e.g. avoidance of dangerous operational regions) or civilian (physical oceanography, study of the evolution of biological species,...). In the past, we assessed the problem of boundary tracking using visual information [1]. However, use of video data in the ocean can be often compromised by lack of ambient light, or by water turbulence. A more robust alternative is the use of acoustic sensors. In this communication, we present work on the definition of signal processing and automatic control algorithms to implement a contour-tracking behavior based on the information provided by a mechanically scanning profiler sonar.

The paper is organized as follows. The next section briefly describes the hardware and software architectures of the platform used for this study. The two subsequent sections, III and IV, are dedicated to the sonar signal processing and to controller definition, respectively. Finally, section V presents results of processing of real data acquired at sea, and simulations of the control structure which is proposed. Section VI summarizes our conclusions, presenting some directions for future improvements.

II. ARCHITECTURE

The underwater platform used in this study is the ROV Phantom,[1] shown in Figure 2. This robot is equipped of three thrusters, two allowing control in the horizontal plane (forward, reverse, turning) and another controlling the motion in the vertical plane (up/down motions), and of the following navigation and perception sensors: a magnetic compass, a rate gyro, a pressure (depth) gauge, an altimeter, a profiler sonar mounted on a tilt platform and a video camera. Moreover, each axis has been equipped of sensors allowing the measurement of the rotation speed of the corresponding motor shaft. The vehicle is linked to a dry-end operational station through an umbilical cable of about 120 meters, which allows remote automatic control of the robot.

On the dry-end, the software architecture of the complete system is spread over three distinct processors. On two personal computers run several threads dedicated to signal acquisition, high-level control and signal processing. Low-level control loops (the motor controllers, as well as basic heading, rate, depth, and altitude control loops) run in a specialized board, which accepts reference values from the other two processors.

The Phantom is programmed through a specially designed user interface, which allows the definition of a mission as a sequence of basic parametrized tasks: go-to, visit a sequence of way-points, visual tracking and station keeping. The actual implementation of all these basic navigation and observation behaviours is the result of the European project NARVAL which finished in 2001, and which was done in collaboration with other partners[2]. This paper concerns the definition of an additional behaviour: sonar tracking, i.e., the use of sonar information to guide an autonomous vehicle along the boundary between distinct habitats occupying the sea floor.

The implementation of this behaviour, done in the context of the programme of the European research project SUMARE, required the definition of two new software modules: the sonar classifier, whose goal is to

[1] Phantom is a Remotely Operated Vehicle produced by Deep Ocean Engineering, USA, which has been made available for research in underwater robotics at I3S through a special educational arrangement.

[2] NARVAL (*Navigation of Autonomous Robots Via Active Environmental Perception*) was an ESPRIT-LTR project, partially funded by the European Community, whose leader was Instituto Superior Técnico. For more information, visit http://isr.ist.utl.pt/

associate a label (class) C_k to each newly acquired profile p_k, and the acoustic controller, whose responsibilty is to generate the command signals r_k that guide the robot along the contour between the distinct classes. The next two sections describe each of these two blocks, shown in bold in Figure 1 below.

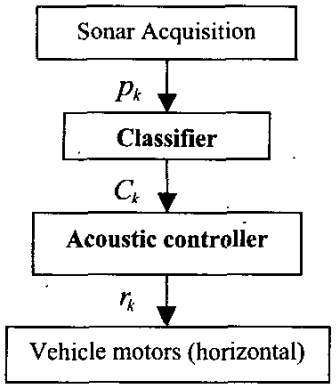

Figure1. Signal processing and control architecture

In the diagram above, the output of the Classifier (the input to the contour tracker) are the *labels* C_k of the received profiles. As we discuss below, we drive the controller, instead, with a continuous signal that indicates the *relative percentage of the classes* defining the tracked boundary inside a sliding window extending over the most recently acquired profiles.

III. SONAR CLASSIFICATION

In this section we describe the signal processing algorithms that produce the relevant information for the contour tracking task. We first describe the sensing equipment on which the work is based, in subsection *A*. Subsection *B* addresses the problem of (unsupervised) learning of the probabilistic models of the observed data on which the classification algorithm presented in subsection *C* is based.

A The sensor

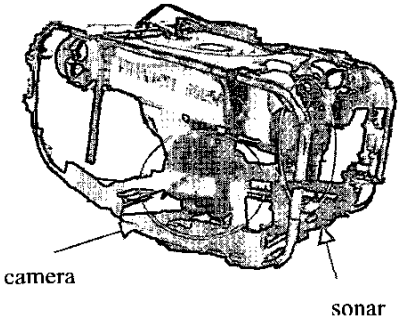

Figure 2: The ROV Phantom and its perception sensors.

The sensor used to scan the sea bed is a dual frequency Tritech Seaking profiler Sonar, mounted the in the ROV Phantom. During our experimentation, the following configuration has been used :

- the sonar is mounted in the front of the ROV (see Figure 2), oriented toward the sea bottom, and scanning the angular sector between +30° and –30°,
- mechanical step size: 0.9°,
- depth (range) resolution: 0.04 m,
- frequency of the emitted signal: 1.2 MHz,
- beamwidth: 1.4° (conical).

B Learning the classes models

The unsupervised segmentation algorithm exploits the fact that the sonar profiles corresponding to sea floor regions occupied by distinct species have *distinct shapes*. Each complete profile p_k is first reduced to a small set of features $\{f_\ell^k\}_{\ell=1}^L$. For the experiments presented in this paper, $L=1$, simply the energy of the profile around the detected maximum (see Figure 3):

$$E_n = \frac{1}{32}\sum_{i=j-16}^{j+15} x_i^2 . \qquad (1)$$

The segmentation algorithm is based on a probabilistic framework, and associates to each possible class C_i a probability distribution of the extracted features, $p\left(\{f_\ell^k\}_{\ell=1}^L \middle| C_i\right)$, $i=1,2$. These probability distributions are initially unknown, and are learned dynamically by the algorithm described below.

We introduce first some nomenclature and notation. Let X be a discrete random variable (rv) with probability space (Ω, A, P) where $\Omega=\{a_1,a_2,\cdots,a_M\}$, is the (finite discrete) realization space, A is a sigma-field of subsets of Ω and P is a probability measure. We denote by lower-case letters x the realizations of X. Consider a sequence $x^{(N)}=\{x_1,x_2,..,x_N\}\in\Omega_N$ of N independent realizations of X. The **type** of $x^{(N)}$, which we denote by $\nu_{x^{(N)}}\Omega\rightarrow[0,1]$ is the empirical estimate of the probability distribution (pd) of X, and is given by:

$$\nu_{x^{(N)}}(a_j)=\frac{1}{N}\sum_{i=1}^{N}1_{a_j}(x_i), j=1,..,M \qquad (2)$$

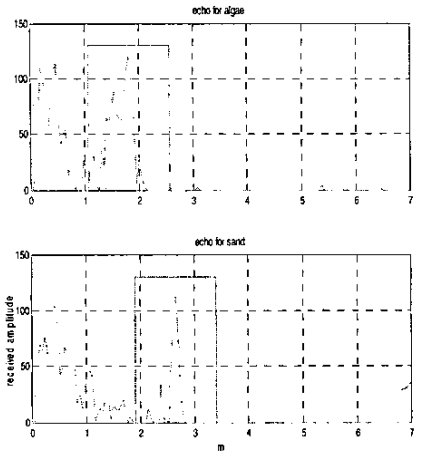

Figure 3: Profiles received from Posidonia (top) and sand (bottom) during one experiment at sea.

where $1_{a_j}(x_i)=\begin{cases}1, & x_i=a_j\\ 0, & x_i\neq a_j\end{cases}$.

Consider that we are given two sequences of length N: $x_1^{(N)}=(x_{21},..,x_{2N})$ and $x_2^{(N)}=(x_{11},..,x_{1N})$. Then, the MDL (Minimum Description Length, see [2]) test for choosing between the two following composite hypotheses:

$$\begin{aligned} &H_0\text{: } x_1^{(N)}\propto {p_0}^N\ x_2^{(N)}\propto {p_0}^N \\ &H_1\text{: } x_1^{(N)}\propto {p_1}^N\ x_2^{(N)}\propto {p_2}^N, p_1^N\neq p_2^N \end{aligned} \quad (3)$$

where the probability laws p_0^N, p_1^N and p_2^N are unknown, i.e., for deciding whether the two sequences were generated by the *same* probability law or if they are samples from *distinct* distributions, is

$$\frac{(M-1)}{M}[2\log(N+1)-\log(2N+1)] \underset{H_1}{\overset{H_0}{\gtrless}} D(\nu\|\hat{\mu})+D(\nu_2\|\hat{\mu}) \quad (4)$$

In the previous expression, $\hat{\mu}$ is the balanced mixture of the types of the two observed sequences, and $D(\|\cdot)$ is the Kullback-Leibler divergence between probability laws:

$$\hat{\mu}=\frac{1}{2}(\nu_1+\nu_2), \text{ and } D(\nu\|\mu)=\sum_{j=1}^{M}\nu(a_j)\ln\frac{\nu(a_j)}{\mu(a_j)}, \quad (5)$$

where ν_1, ν_2 are the types of the sequences $x_1^{(N)}, x_2^{(N)}$, respectively.

Eqs. (4) and (5) show that under the hypothesis that the individual samples (in our case, the set of features extracted from each profile) are statistically independent, the types of the observed sequences are sufficient statistics for the decision problem formulated above.

To learn the classes models, we initialize the probability law of the first class with the type of the sequence of the first N measures: $\hat{p}_1^N=\nu_1$. We then use test (4) to decide if the types of the subsequent sequences, $\nu_k, k>1$, correspond to the same distribution ($\hat{p}_1^N$), until hypothesis H_1 is accepted, for a given $k=k^*$. In this case, the learning phase is then stopped and we set the estimate of the second class probability law equal to the corresponding type: $\hat{p}_2^N=\nu_{k^*}$.

C Classification/Estimation (tracking) phase

We assume in this step that the probability laws $p\left(\{f_\ell^k\}_{\ell=1}^L\middle|C_i\right)$ associated to each class have been learned in a previous step. Consider a set of N consecutive profiles acquired by the robot, $p^k=\{p_k,p_{k-1},...p_{k-N+1}\}$ and denote by f^k the corresponding set of $N\times L$ extracted features.

If during the acquisition of all these N profiles the robot observed the same sea-bed type, then, according to our hypothesis, the type of the sequence f^k should be close (in the Kullback-Leibler "metric") to the corresponding probability law, and the optimal test to decide on the correct class would choose the class m^* that minimizes the Kullback-Leibler divergence between the type of the observed sequence, ν_k and the classes' representatives, p_1^n and p_2^n:

$$m_k^*=\arg\min_{m=1,2} D(\nu_k\|p_m^n). \quad (6)$$

However, the assumption that the observed class is constant during the set of N consecutive profiles is not realistic, and the test above too simplistic. In general, the N successive sonar beams will hit sea-bed regions occupied by distinct habitats, such that a more realistic model for the type of the sequence of length N observed at time k is a *mixture* of the two basic *pd* corresponding to each of the two classes present:

$$\nu_k=\pi_k p_1^n+(1-\pi_k)p_2^n\equiv p^n(\pi_k), \quad \pi_k\in[0,1],$$

where we defined the notation $p^n(\pi)$. Te unkown mixture coefficient π_k indicates *the relative percentage of the two classes* in the observed sequence. The maximum likelihood estimate of π_k is obtained by solving the following minimization problem:

$$\hat{\pi}_k=\arg\min_{\pi} D(\nu_k\|\nu^n(\pi)). \quad (7)$$

We have verified numerically, for a large number of class models, that the criterion $D(\nu_k\|\nu^n(\pi))$ presents a dependency on π which is well approximated by a quadratic:

$$D(\nu_k\|\nu^n(\pi))\cong c_0^k+c_1^k\pi+c_2^k\pi^2, \quad (8)$$

where the unknown coefficients $\{c_i^k\}_{i=0}^2$ depend on the observed type and on the classes models. If these coefficients were known, $\hat{\pi}_k$ could be determined by the following analytical expression:

$$\hat{\pi}_k = -\frac{c_1^k}{2c_2^k}, \tag{9}$$

if the minimum is inside the interval [0,1], and on one of its extrema (0 or 1) otherwise, indicating in this last two cases a "pure type".

In order estimate the coefficients $\{c_i^k\}_{i=0}^2$, we evaluate the criterion $D\left(\nu_k \| \nu''(\pi)\right)$ for three distinct values of π: 0.25, 0.5 and 0.75, and use Eq. (9) to estimate the relative frequency of the classes present in the N most recently acquired profiles.

Note that the estimates $\hat{\pi}_k$ can also be used to establish a *hard segmentation* of the observed sea-bed regions, by using the following simple test:

$$\hat{\pi}_k \underset{\text{class2}}{\overset{\text{class1}}{\gtrless}} 0.5. \tag{10}$$

IV. CONTOUR TRACKING

The role of the acoustic tracker is to use the information yield by the sonar classification algorithm to generate control signals for the robot lower level control loops. To minimize problems due to variability induced by changing observation conditions, we impose that the detected contours be observed at *constant altitude*. This reduces the control problem to guidance in the horizontal plane. Ideally, we want the robot's center of mass to describe a curve that is the parallel translation of the observed contour, whose shape is unknown. We have assessed the problem of contour tracking with an autonomous platform in [3], where it is shown that a classic proportional-derivative controller with suitably defined gains, and using an error signal that is the distance of the robot to the tracked contour in the direction orthogonal to the tracked line, achieves the control objective of driving this distance to zero. In the application considered in [3], tracking of iso-depth lines with a platform equipped of a single beam acoustic altimeter, this error signal can be well approximated by the difference between the altitude of the tracked line and the measured altitude, enabling definition of a robust and completely sensor-driven control approach, not requiring determination of the robot's spatial position.

The problem assessed here presents an additional difficulty: the "sensor" through which the contour is detected is in this case the output of the "Classification/estimation" algorithm presented in the previous section, i.e., the estimates of the mixture coefficients $\hat{\pi}_k$.

Compared to the previous contour tracking problem, which uses altitude measures, this virtual sensor

1. inherently introduces a smoothing of the distance information which should be ideally used to drive the tracking controller;
2. presents hard saturations, for large values of the distance of the robot with respect to the boundary line (corresponding to $\hat{\pi}_k = 0$ or 1).

Convenient design of control loops around sensors with saturations (controllers that typically switch between their own saturation levels, along transition hysthereses) and filtering effects cannot be done using simple PID-like methods, and are, moreover, prone to oscillatory behaviors.

The second problem mentioned above can be minimized by carefully designing the controllers such that the probability of entering the saturation regions is small.

We use the ability of the sonar head to mechanically steer the sonar beam to overcome the first difficulty. We motivate our approach below. Consider that the sonar is made to periodically scan an angular sector $[-\Delta\phi, \Delta\phi]$ with angular resolution $\delta\phi = \Delta\phi / N_s$, in the vertical plane that passes through the center of sonar reference frame (orthogonal to the robot direction of motion). Consider that in the processing algorithm presented in the previous section, we use a sequence length which is equal to the period of the sonar scanning angle, $2N_s$. In this circumnstances, and in the ideal situation when the robot is perfectly aligned with the boundary (the center of the sonar reference frame is at the vertical of the contour) $\hat{\pi}_k$ will be constant and equal to 0.5. Departures form this value indicate an offset of the robot with respect to the boundary. We propose to use the "error" signal

$$e_k \equiv \hat{\pi}_k - 0.5 \in [-0.5, 0, 5],$$

as a substitute for the continuous error distance that drives the iso-depth line tracker in [3].

As it is fully motivated in [3], the natural control input to track an horizontal line is the *yaw rate* of the vehicle, which effectively controls the curvature of the robot trajectory at each point. The output of our acoustic controller is thus the input of the yaw rate controller, imposing an instantaneous rotation speed of the platform in the horizontal plane:

$$r_k = K_p e_k + K_d \left(e_k - e_{k-1}\right), \tag{11}$$

And where the sign of the proportional and derivative gains K_p and K_d depends on whether class 1 is on the left or on the rigth side of the boundary during tracking.

Note that basing the tracking task on this error signal, which is based on the set of $2N_s$ most recently received

sonar profiles, improves the robustness of the algorithm with respect to outliers in the acquired signals. This controller is completely driven by the perception-derived error signal e_k, and thus independent of the assumed dynamic and kinematic platform models, increasing its robustness with respect to modeling errors.

V. EXPERIMENTAL RESULTS

We present below preliminary results using the algorithms presented in the previous sections.

A Segmentation

We present in this section results on real data acquired during real experiments performed at Villefranche-sur-mer (South of France). The regions of the sea bed observed during these experiments are, in its vast majority, either sandy, or occupied by patches of *Posidonia*. Our goal is to detect, using the sonar returns, the boundaries of these patches (which can be neatly observed in the images simultaneously acquired by the Phantom video camera), and use this information to automatically guide the robot along them. During all the experiments the sonar configuration presented in III.*A* has been used. The Phantom has been manually driven (using the joy-stick

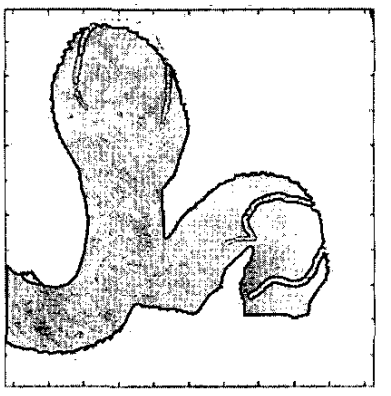

Figure 4: detail of Figure 3.

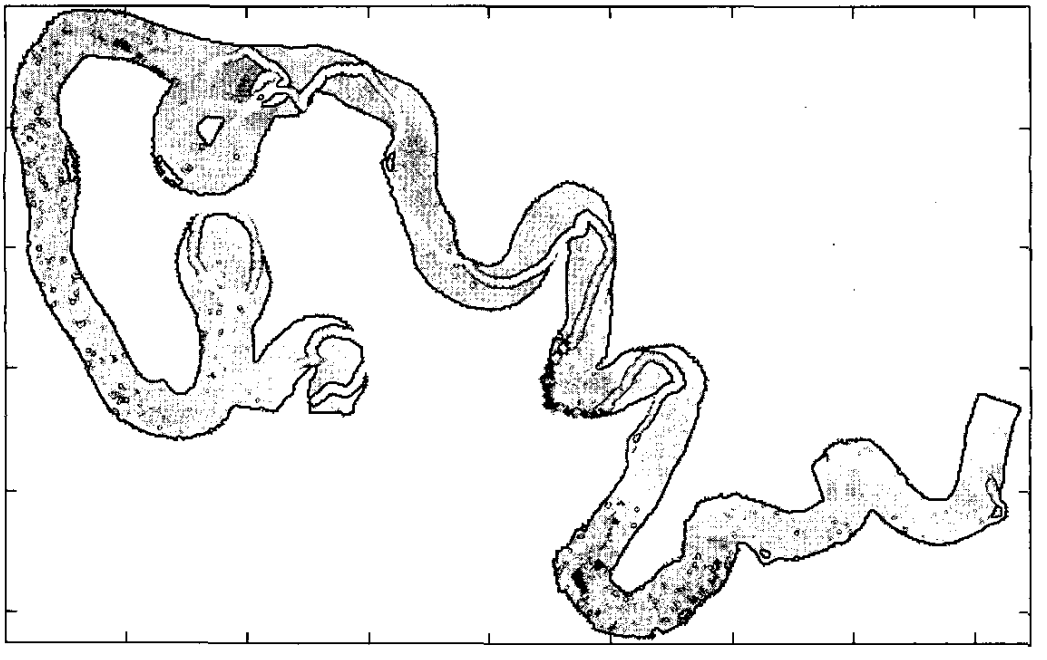

Figure 3: estimated percentage of *Posidonia* along a trajectory.

controls) along a trajectory that crosses several times the boundary of one Posidonia patch, while sonar scans and video images (at a rate of 2 images/sec) were simultaneously recorded.

A mosaic of the acquired video frames has been created off-line, using a correlation method. This video mosaic is used as a ground truth against which the result of the sonar data processing is compared. The result is displayed in Figure 3, that illustrates the performance of our classification/estimation algorithm. Figure 4 is a zoom on a small area, allowing a better appreciation of the agreement between the result of the sonar data processing and the mosaic.

The sonar scans are processed using the method

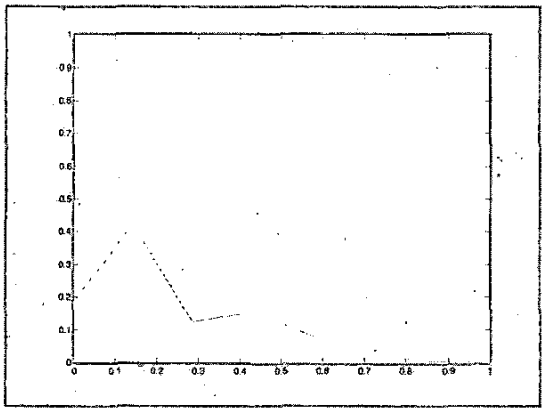

Figure 5: estimated classes' distributions.

presented in section **III**. Sonar scans too close to the sea bottom (profile maximum occurs at a distance less than 0.4m) are not processed, since they correspond to retuns from the ROV crash frame. They are represented in black in the figures.

We first learned, using the algorithm outlined in section III.B, the probability distributions of the two classes (sand and *Posidonia*). The resulting probability laws are shown in yellow (sand) and green (*Posidonia*) in Figure 5.

Using the geometrical model that relates the origins of the visual and acoustic sensors we co-registered the profiles classification onto the video mosaic, using the following color code: white indicates pure sand ($\hat{\pi}_k=0$), blue indicates pure *Posidonia* ($\hat{\pi}_k=1$), while intermediate values are coded by varying intensities of blue. The result is shown in Figure 3, for an observation window of length equal to twice the length of the sonar scans (in the configuration of this experiments this leads to *N=134*), as explained in section IV.

Classification performance is evaluated by comparing the ground truth (provided by the video mosaic) to the output of the hard classification algorithm mentioned at the end of section III.B, and is summarized by the confusion matrix shown in Table 1. For classification purposes, a smaller window of size 32 has been used, the result of the classification being associated to the window's central profile, and yielding the segmentation shown in Figure 6. In this Figure, yellow indicates sand, and green corresponds to *Posidonia*. The red line indicates the trajectory of the robot's centre of mass. It can be seen that the percentage of correct classification is high for both classes, being slightly better for sand than for *Posidonia*. We stress that these numbers give only an indicative idea

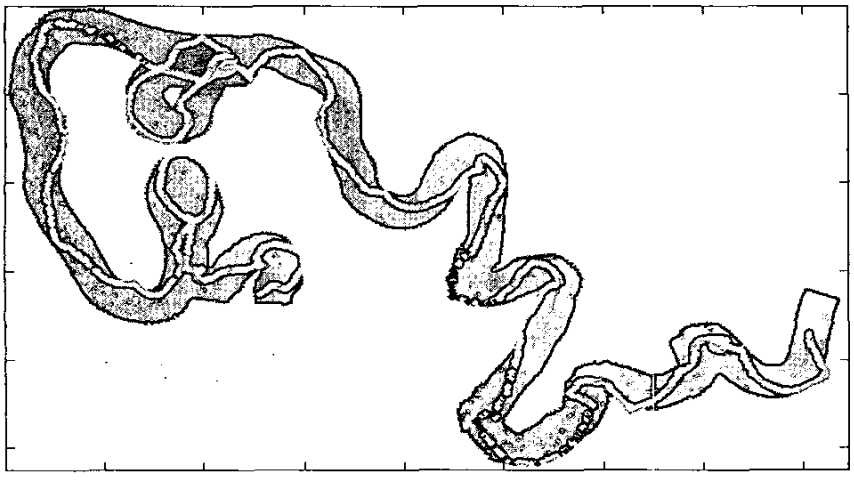

Figure 6: classification (*N=32*).

of the algorithms behavior, being prone to errors in the co-registration process.

TABLE1. CLASSIFICATION RESULTS

	Sand	*Posidonia*
Sand	**97.2 %**	5.8 %
Posidonia	2.8 %	**94.2 %**

B Contour Tracking

We performed simulations of the tracker described in the previous section, considering complete simulation of the motion in the horizontal plane (including thruster dynamics and low level control loops). The robot is driven at a nominal surge speed of 0.5 m/s, control rate is 10 Hz, and sonar rate (rate of classified sonar profiles) is 20 Hz (corresponding to an operating height of 2 meters). Sonar scanning parameters are $\Delta\phi = -5$ and $N_s = 12$, corresponding to an angular cone of $\pm 30°$. Controller gains are $K_s = 0.01$ and $K_D = 2.0$.

Figure 7 shows tracking under noiseless conditions (all profiles are correctly classified) of a contour of small curvature. The top plot shows the tracked contour line (solid black line) and the robot trajectory (in blue). The probed sea-floor points are shown in red. The central plot shows the true class C_k of the hit points, which, as we see, oscillates between –1 and 1, indicating that the robot is well centered above the contour. Finally, the bottom plot shows evolution of twice the signal e_k defined in section IV. As we see, this signal oscillates around zero (the situation of perfect tracking). We plot in Figure 8 the estimate of the mixture coefficient π_k for the experiment shown in Figure 4.

Figure 9 shows tracking of a contour line under simulated classification errors. The probability of error is the same for both classes, and is equal to 0.1. We can see the more erratic aspect of the estimated class signal, C_k, and the irregular structure of the error signal $2\,e_k$.

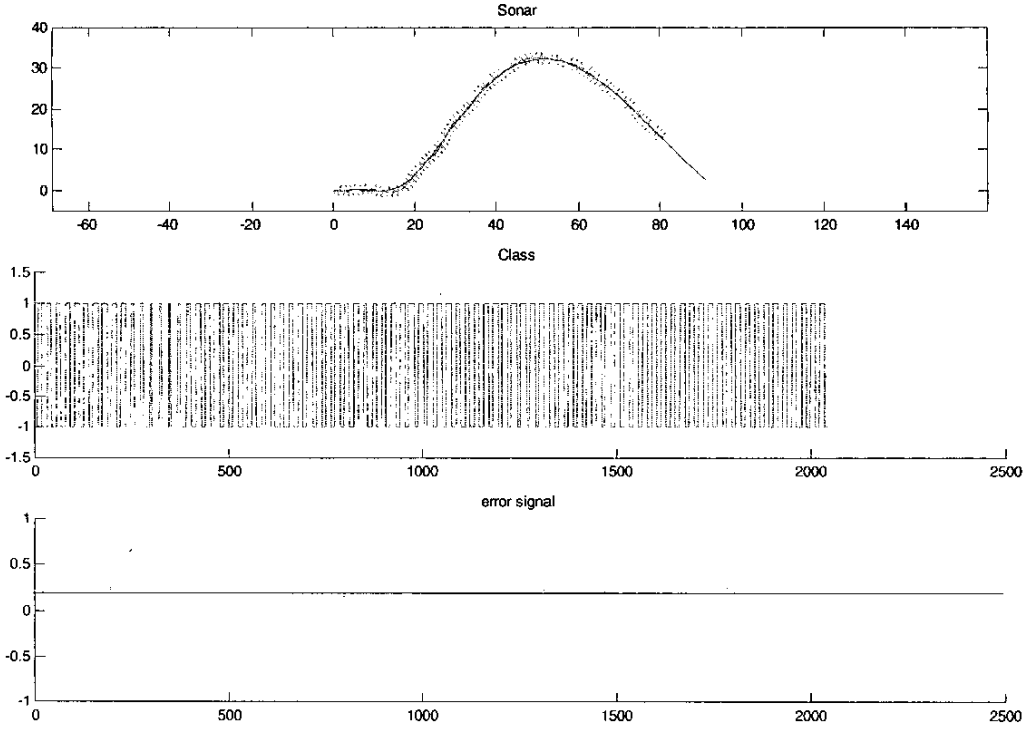

Figure 7: contour tracking (noisless).

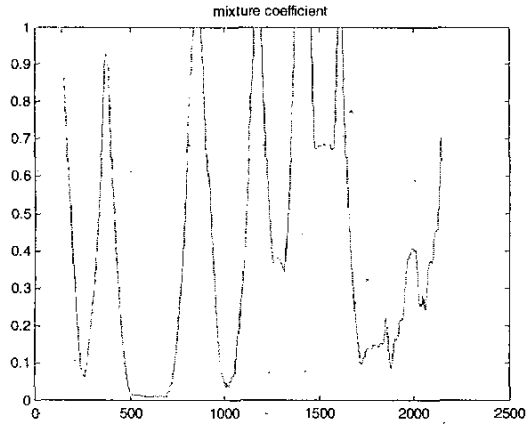

Figure 8: estimated mixture coefficient along trajectory of Figure 4.

VI. CONCLUSIONS

We presented algorithms for contour tracking using data provided by a profiler sonar. The main innovation of the approach is the proposition of basing tracking control on the estimation of the parameters of a mixture model that is fit to the most recently received sonar returns. Results of this data processing algorithm on real data collected at sea for the boundary between two natural sea bed types demonstrate the adequancy of the unsupervised segmentation presented in the paper. A convenient error signal is obtained by centering the estimated mixture coefficient, which drives a simple discrete classical proportional-derivative controller.

Several future directions for improving performance are currently under study. One consists in considering simultaneous control of the robot's yaw rate *and* surge speed. Indeed, when considering constant surge speed, controlling yaw rate is equivalent to controlling the curvature of the robot's trajectory, which is equal to the ratio $k=r/u$, where u is the surge speed. Since the robot's rotational speed is bounded in absolute value, it may happen that to attain the desired rotational speed, it is necessary to decrease surge speed.

ACKNOWLEDGMENTS

This work has been partially funded by the European Union through the IST programm, project SUMARE (*Survey of Marine Resources*), contract IST-1999-10836. See **http:/www.mumm.ac.be/SUMARE** for more details.

REFERENCES

[1]. Albert Tenas, Maria-João Rendas, Jean-Pierre Folcher, *Image Segmentation by Unsupervised Adaptive Clustering in the Distribution Space for AUV guidance along sea-bed boundaries using Vision*, Proc. OCEANS 2001, Honolulu, Hawaii, USA, November 2001.

[2]. Jorma Rissanen, *Stochastic Complexity in Statistical Inquiry*, World Scientific, Series in Computer Science—Vol. 15, 1989.

[3]. Maria-João Rendas, Isabel Lourtie, Georges Pichot, *Adaptive Sampling for sand bank mapping using an autonomous underwater vehicle equipped of an altimeter*, ISESS 2003, Vienna, Austria, May 2003.

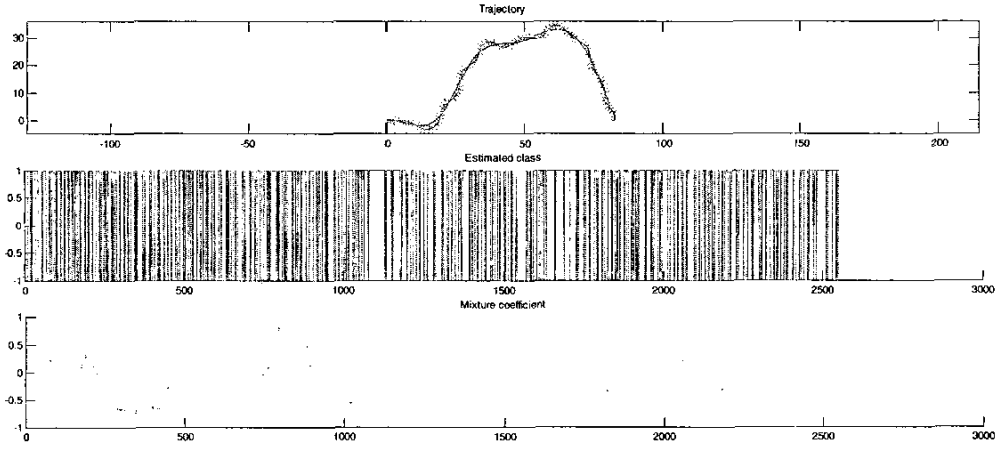

Figure 9: contour tracking (noise, error probability = 0.1).