

# Benthic boundary tracking using a profiler sonar

**Christian Barat**

Laboratoire I3S (Informatique, Signaux et Systèmes de  
Sophia Antipolis), CNRS-UNSA  
2000 route des Lucioles, BP 121, 06903 Sophia Antipolis  
cedex, FRANCE

[barat@i3s.unice.fr](mailto:barat@i3s.unice.fr)

**Maria João Rendas**

Laboratoire I3S (Informatique, Signaux et Systèmes de  
Sophia Antipolis), CNRS-UNSA  
2000 route des Lucioles, BP 121, 06903 Sophia Antipolis  
cedex, FRANCE

[rendas@i3s.unice.fr](mailto:rendas@i3s.unice.fr)

## Abstract

*The paper presents signal processing and control algorithms that enable autonomous tracking of the 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 profilers to discriminate between distinct materials. By exploiting sonar scans of the region below the robot, a classical control loop is closed around the sonar classifications, using a feedback signal that is robust with respect to classification errors'.*

## Keywords

*Contour tracking, unsupervised classification, underwater robotics.*

## INTRODUCTION

The ability to track natural boundaries defined in the ocean floor by distinct habitats is useful in several applications, either military (avoidance of dangerous operational regions) or civilian (physical oceanography studies, 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.

## ARCHITECTURE

The underwater platform used in the study is the ROV Phantom,<sup>1</sup> 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.

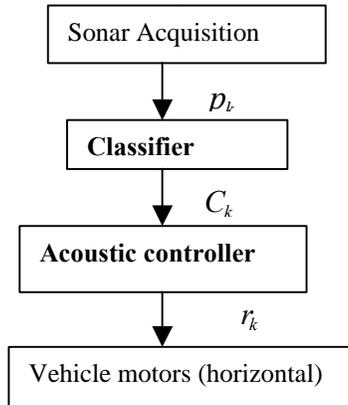
The software architecture of the complete system is spread over three distinct processors. On two personal computers run several threads that are dedicated to signal acquisition, high-level control and signal processing. Low-level control loops (the motor controllers, as well as basic heading, rate, depth, and altitude control loops) run in a specialized board, which accepts reference values from the two other 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 was done in collaboration with other partners<sup>2</sup>. 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.

<sup>1</sup> 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.

<sup>2</sup> 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>

The implementation of this behaviour required the definition of two new software modules: the sonar classifier, whose goal is to associate a label  $C_k$  (class) to each newly acquired profile  $p_k$ , and the acoustic controller, whose responsibility is to generate appropriate commands  $r_k$  that guide the robot along the contour between the distinct classes. The next two sections describe each of these two blocks, in bold in the Figure below.



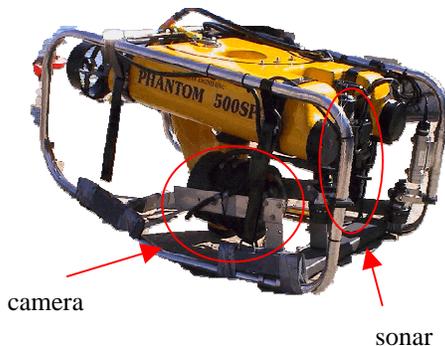
**Figure1. Signal processing and control architecture**

**SONAR CLASSIFICATION**

**The sensor**

The sensor used to scan the sea bed is a Tritech Seaking profiler Sonar. During our experiments, the following configuration has been used :

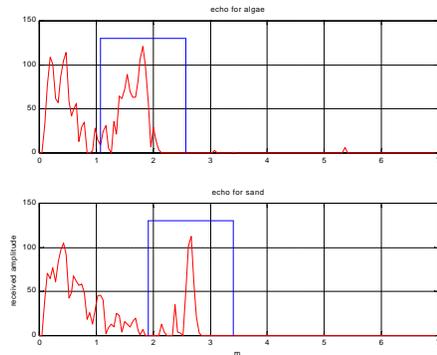
- The sonar is mounted in the front of the ROV (see Figure 2). It is oriented towards the sea bottom and scans between +30° to mechanical step 0.9°.
- Depth resolution: ± 0.02 m.
- Frequency of the emitted signal: 1.2 MHz.
- Beamwidth: 1.4° Conical



**Figure 2: The ROV and the exteroceptive sensors.**

**The features**

The sea bottom observed at Villefranche-sur-mer presents only two classes: posidonia and sand. To perform pattern recognition, features must be extracted from the raw data.



**Figure 3: Echoes received from posidonia and sand sea bottom.**

**Method**

The segmentation algorithm exploits the fact that the sonar profiles corresponding to sea floor regions occupied by distinct materials (in our experiments we work in a boundary between sand and Posidonia) have *distinct shapes*. Discriminative features have been selected to describe each received profile, such that each complete profile  $p_k$  is reduced to a small set of parameters  $f_k$ . In the current implementation  $f_k$  is simply the energy of the profile around the detected maximum (see eq. 1).

$$E_n = \frac{1}{32} \sum_{i=j-16}^{j+15} x_i^2 \tag{1}$$

The segmentation algorithm associates to each possible class  $C_i$  a probability distribution of these features,  $p(f_k|C_i)$ , which is initially unknown. These probability distributions are learned dynamically by the algorithm described below. We first introduce some nomenclature and notation. Let  $X$  be a discrete random variable (rv) with probability space  $(\Omega, A, P)$  where  $\Omega = \{a_1, a_2, \dots, a_L\}$  is the (finite discrete) realization space,  $A$  is a sigma-field of subsets of  $\Omega$  and  $P$  is a probability measure. We denote by

lower-case letters  $x$  the realizations of  $X$ . Consider a sequence  $x^{(n)} = \{x_1, x_2, \dots, x_n\} \in \Omega^n$  of  $n$  independent realizations of  $X$ . The **type** of  $x^{(n)}$ , which we denote by  $\mathbf{n}_{x^{(n)}} : \Omega \mapsto [0,1]$  is the empirical estimate of the probability law of  $X$ , and is given by:

$$v_{x^{(n)}}(a_j) = \frac{1}{n} \sum_{i=1}^n 1_{a_j}(x_i), j=1, \dots, L, \quad (2)$$

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

Consider that we are given two sequences of length  $n$ :

$\mathbf{x}_1^{(n)} = (x_{11}, \dots, x_{1n})$  and  $\mathbf{x}_2^{(n)} = (x_{21}, \dots, x_{2n})$ , of iid discrete  $rv$ 's taking values in alphabet  $\Omega = \{a_1, \dots, a_L\}$ , with  $|\Omega| = L$ . The MDL (Minimum Description Length, see [2]) test for choosing between the two hypothesis:

$$\begin{aligned} H_0 : x_1^{(n)} \propto p_{\mathbf{m}_1}^n \quad x_2^{(n)} \propto p_{\mathbf{m}_2}^n \\ H_1 : x_1^{(n)} \propto p_{\mathbf{m}_1}^n \quad x_2^{(n)} \propto p_{\mathbf{m}_2}^n, \mathbf{m}_1 \neq \mathbf{m}_2 \end{aligned} \quad (3)$$

where  $v_1, v_2$  are the types of the sequences  $x_1^{(n)}, x_2^{(n)}$  is:

$$\frac{(L-1)}{L} (2 \log(n+1) - \log(2n+1)) \underset{H_1}{\overset{H_0}{>}} D(\mathbf{n}_1 \| \hat{\mathbf{m}}) + D(\mathbf{n}_2 \| \hat{\mathbf{m}})$$

where

$$\hat{\mathbf{m}} = \frac{1}{2} (\mathbf{n}_1 + \mathbf{n}_2), \quad (4)$$

$$D(\mathbf{n} \| \mathbf{m}) = \sum_{j=1}^L v(a_j) \ln \frac{\mathbf{n}(a_j)}{\mathbf{m}(a_j)}.$$

The algorithm is composed of two steps: a *learning* phase, on which the platform autonomously learns the probability distributions that describe the classes present using the expressions given above, and a subsequent *tracking* phase during which the Bayesian test presented below uses the probabilistic model learned in the previous stage to assign a label  $C_k$  to each received profile.

**Classification**

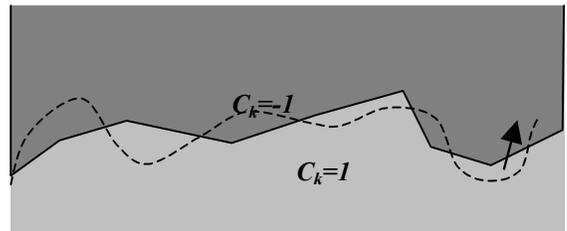
Association of each type  $v_i$  to a class  $k$  is made by choosing the class that minimizes the Kullback-Leibler distance to the classes' representatives:

$$D_k^*(\mathbf{n}_i) = \min_m D(\mathbf{n}_i \| \mathbf{n}_m^n). \quad (5)$$

The next section presents the contour tracking method using the segmentation of the sonar measures.

**CONTOUR TRACKING**

The role of the acoustic tracker is to use the information yield by the sonar classification algorithm, i.e., the sequence of indexes  $C_k$ , 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 gravity 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 horizontal 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 is approximated proportional to the difference between the altitude of the tracked line and the measured altitude. The problem assessed here presents an additional difficulty: the sensor through which the contour is detected is discrete, giving only information about which side of the contour the robot is placed on, see Figure 4. In this sense, useful control information is concentrated only in the instants at which the detected class changes, indicating that the robot *crossed* the contour. Convenient design of control loops around discrete sensors (controllers that typically switch between saturation levels, along transition hystereses) cannot be done using simple PID-like methods, and are prone to oscillatory behaviors.



**Figure 4 : Sensor information during acoustic tracking of a contour between two distinct sea bed habitats**

We use the ability of the sonar head to mechanically steer the sonar beam, to base control in a smoothed

signal which allows a convenient level of performance. We briefly explain the methodology below. Consider that the sonar is made to periodically scan an angular sector  $[-\Delta f, \Delta f]$  with an angular resolution  $d\mathbf{f} = \Delta \mathbf{f} / N_s$ , in the vertical plane that passes through the center of sonar reference frame (orthogonal to the robot direction of motion).

Ideally, in the absence of classification errors, and when the robot is perfectly aligned with the boundary (the center of the sonar reference frame is at the vertical of the contour) the detected signal will be a periodic signal that oscillates between +1 and -1, with a period equal to  $N_s$ . The signal

$$e_k = \sum_{n=k-N_s}^k C_n \quad (6)$$

is thus an indicator of the offset of the robot with respect to the boundary: it is equal to zero when the robot is centered on the contour, negative if it is on the left side of the boundary, and positive otherwise. We propose to use this "error" signal as a substitute for the continuous error distance that drives the iso-depth line tracker in [3]. The discrete nature of this error signal prevents consideration of a derivative term in the control law.

As it has been fully motivated in [3], the natural control input to track an horizontal line is to control the yaw rate of the vehicle, which is equivalent to controlling 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:

$$r_k = K_s e_k \quad (7)$$

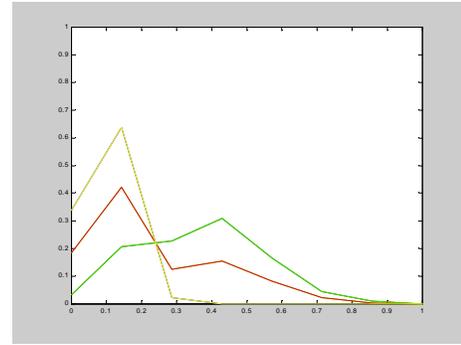
with  $e_k$  defined by eq. (6).

### EXPERIMENTAL RESULTS

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

#### Segmentation

Using the segmentation algorithm we detect 2 classes. The histograms are estimated with a sequence composed of 50 measures (see figure 5).



**Figure 5: Estimated histograms for the 2 classes (yellow = sand and green = posidonia) and the mixture (red).**

Sonar scans and video images (2 images/sec) were recorded along the trajectory of the robot, driven manually along a boundary that crosses posidonia and sand regions. Off line, a mosaic of video image was created to enable evaluation of the classification result. The sonar scans are classified using the segmentation method. Profiles for which the maximum is less than the minimal distance of 0.4m were rejected, since they correspond to returns due to the crash-frame of the robot. This resulted in the elimination of 12.2% of the received profiles. Using the geometrical model that relates the sonar and video frames, we can superimposed the classified scans onto the mosaic images, as shown in figure 6. light points (yellow) correspond to sand, and medium gray (green) to Posidonia. The table below presents the classifier's performance.

**Table1. Classification results**

	class sand	class posidonia
class sand	97.2 %	5.8 %
class posidonia	2.8 %	94.2 %

As Figure 6 shows, the classifier has a stable behaviour, with large sections being assigned to the same class. The majority of the errors contributing to the off-diagonal terms of the previous table are most probably due to misalignment of the sonar and video reference frames, inducing spurious errors in the boundaries.

We do not have yet results of sea trials demonstrating the tracking performance under close loop of the complete system. We hope to be able to present experimental results of the complete system for the final version of the paper, and present below simulation results that illustrate the expected level of tracking performance.

### Contour Tracking

We performed simulations of the tracker described in the previous section, considering complete system simulation of the motion in the horizontal plane (including thruster dynamics and low level control loops). The robot was 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 (operating height is 2 meters). Sonar scanning parameters are  $\Delta f = -5$  and  $N_s = 9$ . Controller gain is  $K_s = 0.01$ .

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 evolution of the classification signal  $C_k$ , 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 the error signal  $e_k$ .

As we see, this signal oscillates around zero (the situation of perfect tracking). Figure 8 shows tracking of a contour with regions of larger curvature, using a scanned cone of larger angular aperture.

In this last case, we can see distinguish an initial acquisition phase, during which the error signal saturates twice, followed by a period of effective tracking of the contour line.

Figures 9 and 10 show tracking of a contour line under classification outliers (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 classification signal  $C_k$ , and the irregular structure of the error signal  $e_k$ . Note the smoothing effect of

the control signal, which is increased in the case of Figure 10, for which a wider scanning angle is used.

### CONCLUSIONS

We presented below preliminary results on contour tracking using classified sonar profiles. 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, at 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. Additionally, we are currently studying the use of more than a single feature to classify sonar returns, still using a non-parametric approach to modeling of the classes' distributions.

### 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.

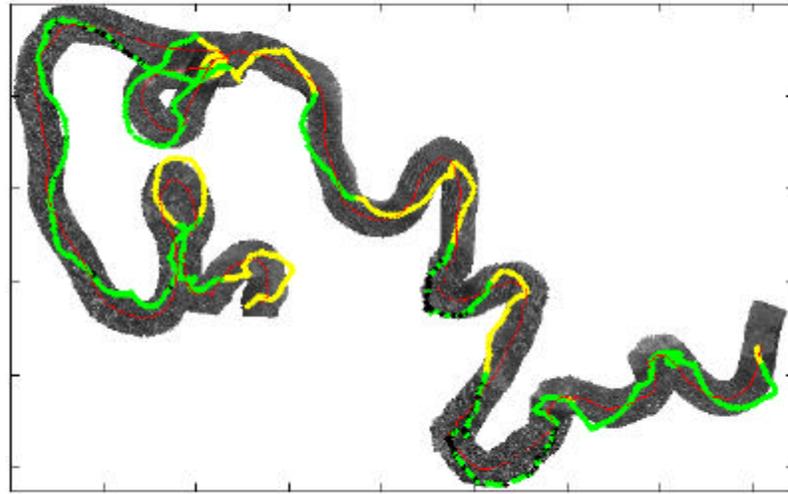


Figure 6: Classification of sonar measures in 2 classes using the segmentation method. In yellow the classification into class sand, in green into the class posidonia.

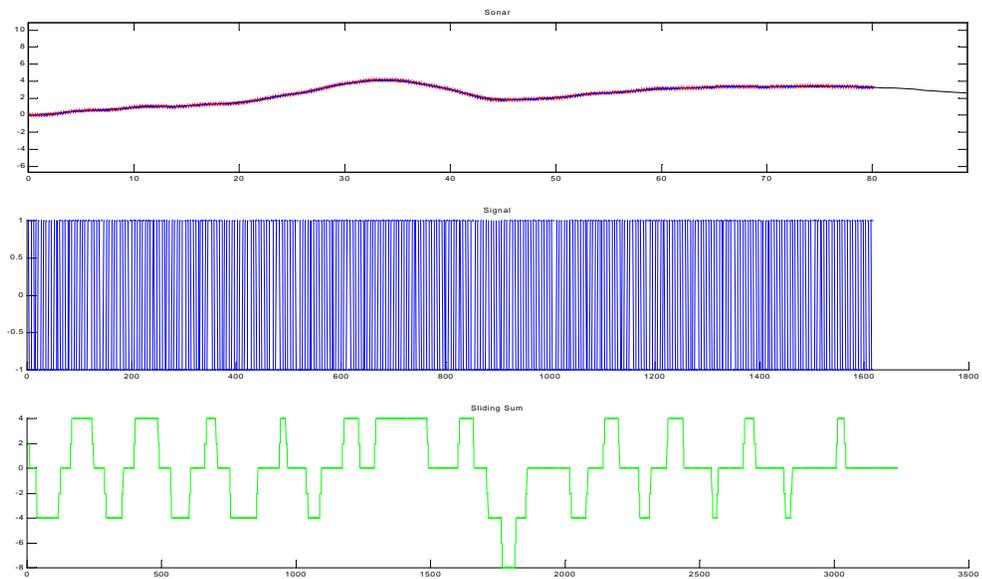


Figure 7: Tracking a contour.

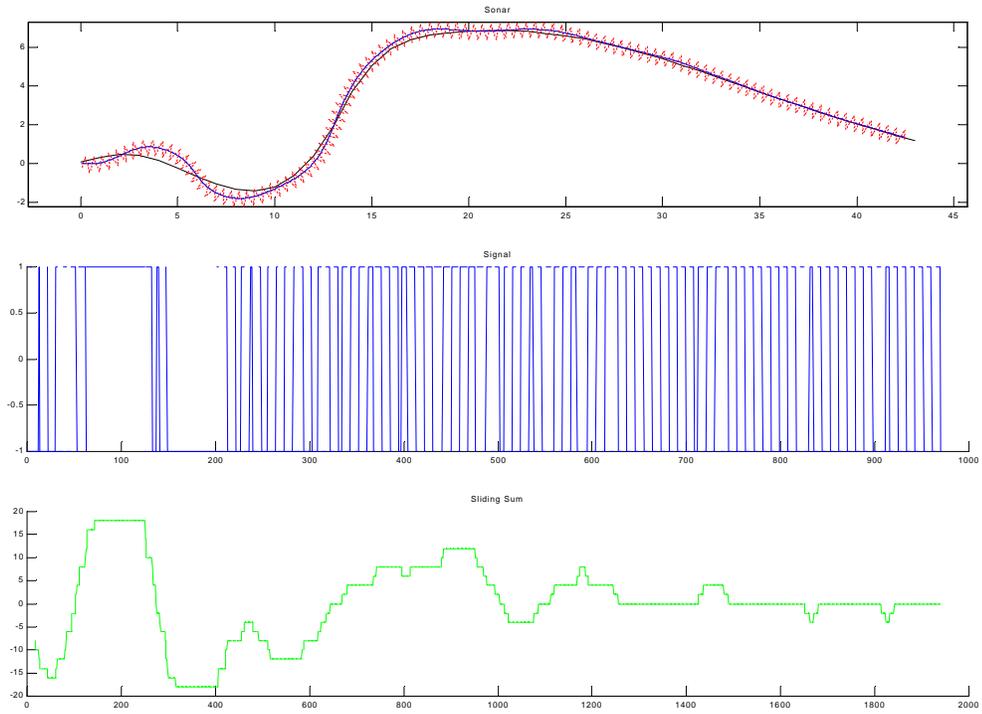


Figure 8: Tracking in noiseless situation.

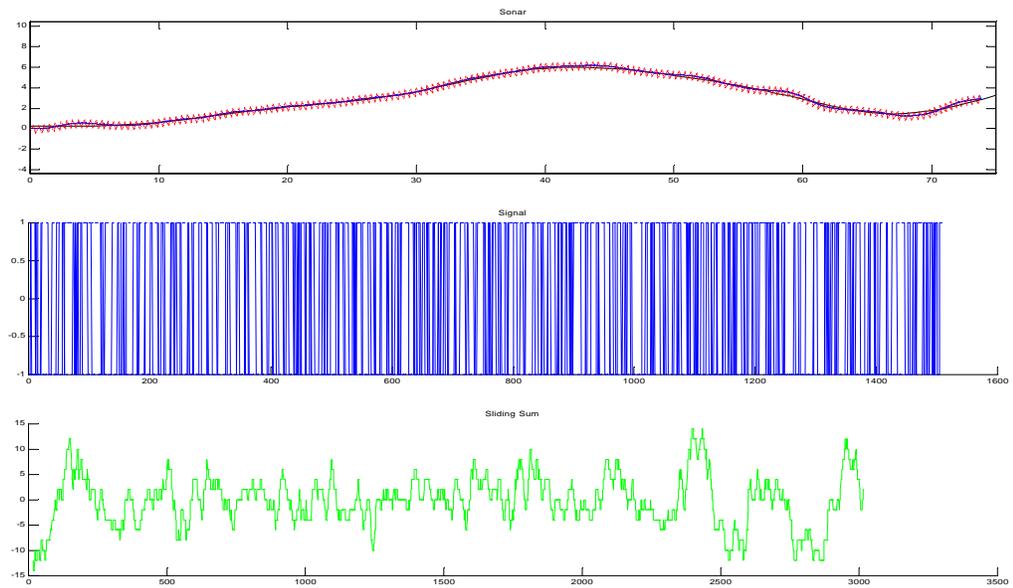
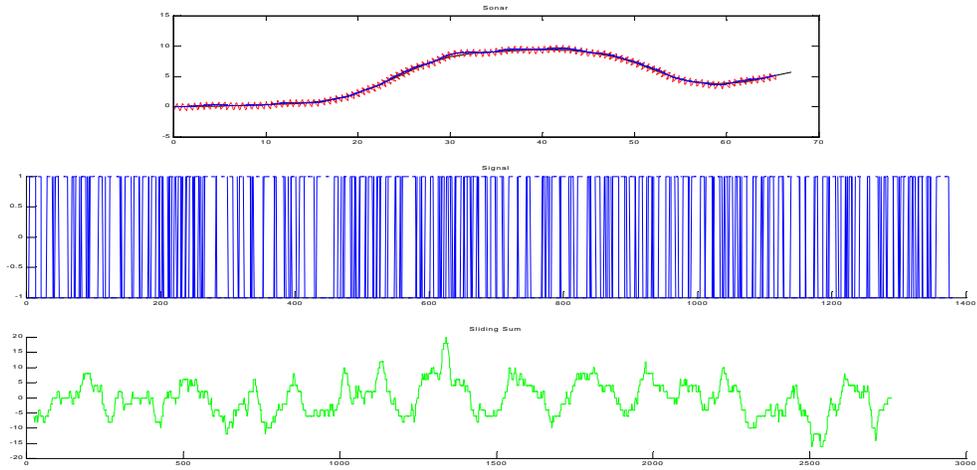


Figure 9: Simulation under classification errors (Probability of error = 0.1).



**Figure 10: Tracking under noise (larger scan, Probability of error 0.1).**