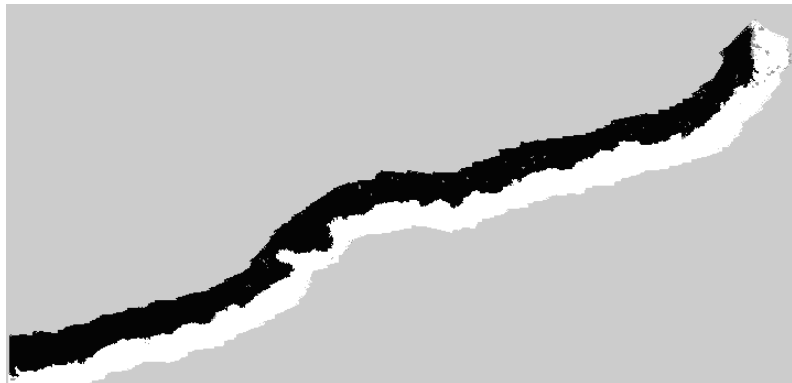

Project SUMARE

Deliverable 4.1 : Contour tracking (with video and altimeter)



Coordinated by: Maria João Rendas

Date: September 2002

Participant partners: I3S (*M.-J. Rendas, J.-P. Folcher*), IST (*I. M. Lourtie*)

Table of Contents

Introduction	3
Overall goals	4
Guidance along contours of distinct benthic regions using vision	4
Guidance along benthic contours using sonar	4
Guidance along iso-depth lines using altitude measures	4
Contour following with video	6
Problem statement	6
State-of-the-art	8
Perception/control architecture (hard contours)	9
Perception/control architecture (fuzzy contours)	9
Contour detection	11
Video segmentation	11
Mosaicing	14
Contour selection	18
Visual Controller	20
Experiments at sea	22
Contour Tracking with Altimeter	30
Problem statement	30
Approach	32
Contour tracker	33
Simulation results	40
Future work	42

Conclusions	44
Bibliography	45

Introduction

This deliverable reports on the work conducted in SUMARE on the definition of one perception-based guidance mode for the two underwater platforms operated in the project: ***contour tracking***.

Hardware problems with the platforms, in particular with the AUV¹ MAUVE, precluded the complete achievement of the work program as planned in the Technical Annex of the project, as it is discussed in detail below. For this reason, this deliverable presents experimental results only on contour tracking using vision, while for contour tracking using acoustic (altimeter) measures, i.e., tracking of iso-depth contour lines, we limit ourselves to the theoretical presentation of the approach that will be implemented and tested as soon as all hardware problems are solved, illustrating its behaviour with simulation results.

The main practical result of the work done until now is the availability of an operational visual contour tracker that has been implemented and tested at sea with the ROV Phantom. The approach chosen by the consortium relies on the use of a novel unsupervised segmentation algorithm, which enables auto-evaluation of the feasibility of this observation mode, given the actual characteristics of the sea-bed. This issue is discussed in full detail in the next chapter. This deliverable is related to two of the project's milestones (Mil 4.1 and 4.2) which concern the ability of implementing on-line guidance algorithms, on the basis of which autonomous observation modes can be defined. Even if the initial goals have not been fully attained (yet), we consider that the positive results obtained so far enable us to be confident that, once the robotic platforms become fully operational, there will be no major problem in implementing the perception guided modes which are not yet available at this time.

The deliverable is organized in 6 separate chapters:

1. *Introduction*. This chapter presents the three distinct perception guidance modes that have been foreseen in the Technical Annex.
2. *Contour tracking with vision*: we present in this chapter the overall structure of the perception/guidance contour tracker that has been implemented in the ROV² Phantom.
3. *Contour detection*. This chapter present briefly the unsupervised image segmentation algorithm that generates the contour information, which has been fully described in Deliverable 3.3.a.

¹ AUV: Autonomous Underwater Vehicle.

² ROV: remotely Operated Vehicle.

4. *Visual Controller*. This chapter presents the design of the control loops that guide the platform along the contour detected on successive image frames, including results of real experiments at sea.
5. *Contour tracking with altimeter*. this chapter presents the approach formulated by the project to the problem of tracking iso-depth lines using an altimeter sensor, illustrating its performance with simulation results.
6. *Conclusions*. This chapter summarizes the main results, and lists issues that will be further studied during the rest of the project.

Overall goals

Three distinct perception-based guidance modes have been identified in the technical Annex of the project : guidance along contours of distinct benthic regions using vision, guidance along benthic contours using sonar, and guidance along iso-depth lines using altitude measures. We define below the behaviors that correspond to each of these modes.

Guidance along contours of distinct benthic regions using vision

In this mode, the platform should autonomously track the limits between regions of homogenous sea-bed occupancy based on the images acquired by a video camera, while producing at the same time a map of its geometry. This goal has been attained, and conclusive experiments with the ROV Phantom that demonstrate the performance achieved are included in this deliverable.

Guidance along benthic contours using sonar

The distinction between this mode and the previous one consists in the sensor that is used to observe the sea-floor, and to detect the boundary line to be tracked. While when using a video camera, an rectangular area of the sea-floor below the ROV is observed at each time instant, the use of acoustic equipment (sonar), only enables the observation of the sea floor along a line. While the work on sonar segmentation has already been started in the project, and preliminary encouraging results are available [BaratRendas2002], the corresponding control system has not yet been designed and tested. This mode will be the issue of separate deliverables, D3.3.b and D4.2, that will report on the segmentation-contour detection and control sub-systems, respectively.

Guidance along iso-depth lines using altitude measures

The goal of this control mode, which is closely related to the problem of producing bathymetry maps for the sand-bank application considered in the project, is to autonomously guide the vehicle MAUVE along iso-depth lines using the data acquired by an altitude sensor (altimeter). The observation structure of this problem is similar to the previous one (only the point in the sea-floor below the robot is observed at each instant). However, an additional information is provided by the *variation* of the altitude data, which is binary for the previous problem, and which can be exploited to obtain a more stable behavior while tracking. Delays in the development of the MAUVE system precluded so far actual implementation and testing of this control mode. In this deliverable, we limit

ourselves to the formal presentation of our approach, relegating the presentation of experimental results to a future deliverable of the project (hopefully, this discussion will be included in deliverable D3.2, otherwise, we will present it in deliverable D6.4).

Contour following with video

This chapter describes the perception/control architecture of the visual contour tracker that has been designed and implemented in the ROV Phantom, integrating signal (image) processing algorithms developed in Work-package 3 (Sensing) and guidance laws defined in Work-package 4 (Guidance). It also describes on-going work on a tracker for guidance along regions of smooth transition between distinct sea-bed habitats.

Problem statement

In the context of underwater robotics, the problem of visual contour tracking can be intuitively formulated in the following way:

Given a robot equipped with a video camera, generate the control signals that *drive the platform along the boundary between distinct benthic regions*.

The formulation above, while intuitively meaningful, does not define a tractable control objective, to which the tools of automatic control theory can be applied. In fact, the two expressions in italic, “*drive the platform along the boundary*” and “*distinct benthic regions*” are not precisely defined.

While we can arbitrarily set a desired tracking mode, imposing that (see Figure 1) “*drive the platform along the boundary*” means “**keep the horizontal distance between the robot’s camera origin and the contour equal to zero and at a constant altitude h above the sea floor**”

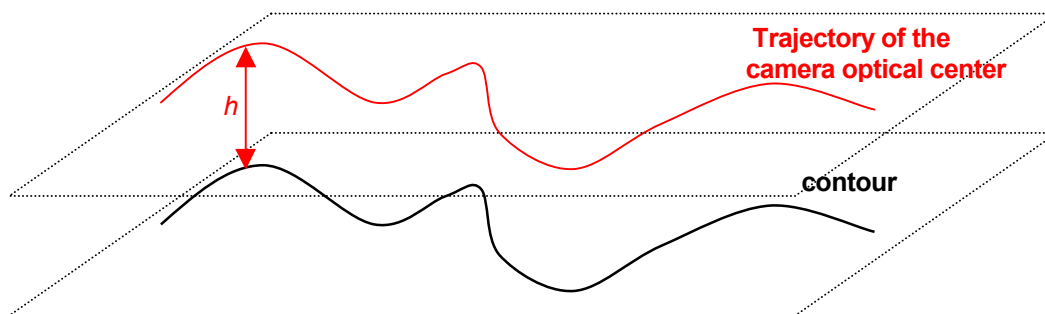


Figure 1: tracking a well defined contour.

as we do, there are many possible ways to “track” a boundary. In fact, any solution that maintains the boundary inside the field of view of the camera is a valid solution to the problem.

Also, for most underwater scenarios, the term “*distinct benthic regions*” is hard to define. Actually, as we will see in subsequent chapters of this deliverable, there is not, in general, a neat boundary (a 1 dimensional manifold) separating clearly distinct sea-bed habitats, but rather a smooth, progressive transition from one kind of habitat to another. This point is discussed in deeper detail on the section on On-line video segmentation. In regions of the sea-bed where it does not exist a clear edge separating two habitats, triggering of naïve contour-tracking algorithms can lead to a chaotic behavior of the platform. For this reason, we have been developing two alternative complementary approaches

- In those regions of the sea-bed where a *clear “edge”* between two adjacent regions is present, we keep the control objective enunciated above;
- When there is a *gradual* transition between two distinct types of benthic occupation, i.e., when the “boundary” is in fact a “transition region”, we adopt the following alternative control objective:

“drive the platform along the boundary” means “keep a constant altitude above the sea floor, while controlling the robot in an oscillating (yo-yo) trajectory covering the transition region”

This alternative control objective is illustrated in Figure 2.

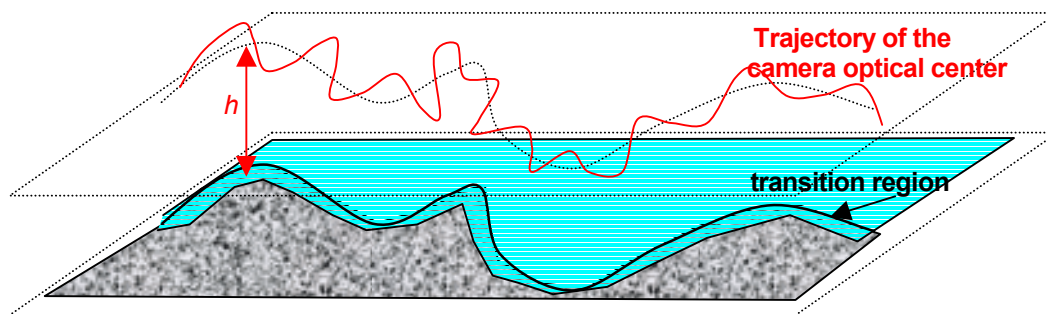


Figure 2: tracking a fuzzy contour.

As the discussion above shows, the autonomous observation behavior of the robot is largely dictated by the characteristics of the sea floor in its operating region. To achieve reliable fully autonomous operating mode, it appears thus that an important problem, which was not acknowledged in our initial intuitive formulation of the contour tracking problem, is

Automatically decide, using the video images of the sea-bottom taken by an autonomous robot, whether

- H_0 : there is no contour in the (locally) observed sea-bed
- H_1 : there is a well defined contour line
- H_2 : there is a detectable transition region of non-negligible width.

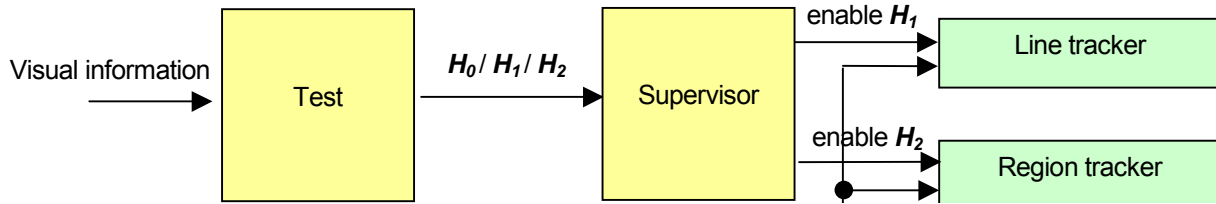


Figure 3: contour tracking guidance.

This **decision** problem is an important component of any contour tracker for an autonomous platform, providing the enabling event that must condition triggering of this mode by the real-time management system of the robot, as shown in Figure 3. Once the operational conditions are correctly evaluated, the corresponding tracking mode can then be safely engaged. In the project, we developed distinct solutions to address each of the two situations described. While the simpler situation of tracking a line contour has been implemented and tested at sea, the “tracker” (“mapper” would be a better designation) for transition regions is still under development, and only initial simulation results are available.

State-of-the-art

Contour tracking has first been studied in the robotics vision community as the problem of shape (or object) tracking: to control the extrinsic parameters of a camera (usually its pan and tilt) in order to maintain a given object inside the camera field of view, assuming that the object’s shape can change, usually according to a set of parametrized deformations. The main difficulty in this problem is caused by the presence of other objects in the observed scene, which can lead to ambiguous association between the distinct contours detected in successive frames. Our problem differs from this tracking problem in that the goal is to control the observer (the robot carrying the camera) in order to perceive the unobserved parts of the object’s contour, and no a priori model for the future observations (the unobserved part of the contour) is in general available. To distinguish the problem addressed here from this one, we will adopt the designation “contour following” instead of contour tracking.

Several teams of underwater robotics have studied the use of visual information for guidance along geometric features of the environment. However, the large majority of these studies concern tracking of man-made structures, such as pipe-lines or cables. Even if these structures may be corrupted by fouling, which may make their recognition difficult, their simple (known) geometry can be exploited to constrain the robot’s motion. To our knowledge, no other operational underwater natural contour tracker besides the one that we developed in the context of the SUMARE project has been reported in the scientific and technical literature.

Existing systems operate in a semi-autonomous mode: the contour (or object to track) is first manually selected by a human operator, who specifies the feature or region to be tracked. The integrated formulation of the problem that has been presented in the previous section, integrating decision tools that indicate which tracker should be used in the current operating conditions is thus another major innovative aspect of the project’s work.

Perception/control architecture (hard contours)

Figure 4 shows the block diagram of the contour tracker for well separated regions, i.e., where a rapid transition between distinct sea-floor regions can be detected (block named “line tracker” in Figure 3).

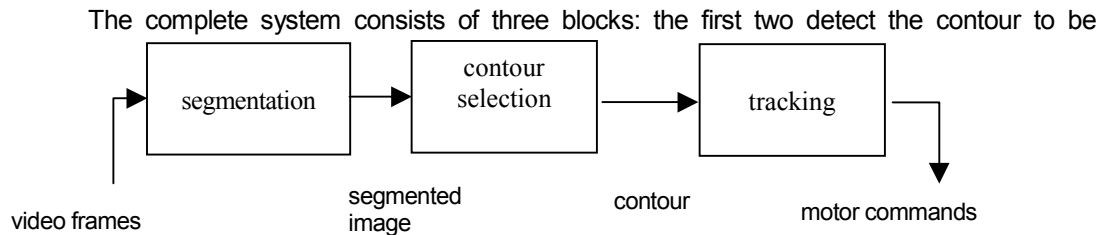


Figure 3: Processing architecture for line tracking.

tracked amongst the boundaries found by an unsupervised image segmentation algorithm (the first block), and the third block (the actual controller) generates the motor commands that keep the robot on top of the detected curve. These blocks are detailed in Chapters 3 (the first two) and 4 of this deliverable.

Perception/control architecture (fuzzy contours)

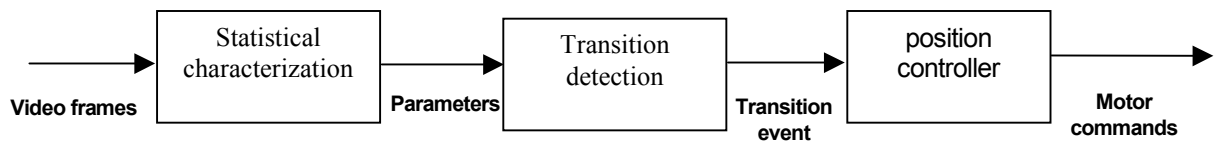


Figure 4: Processing architecture for transition tracking.

Figure 5 presents the block diagram of the architecture for tracking gradual transitions. The video frames are input to a block that fits a random model to the regions found in the video images (by the same unsupervised segmentation algorithm that is used for contour tracking). The parameters of the random model are then fed to a block whose role is to detect significant variations of these parameters. In this case, an event is signaled to the position controller, that should redirect the vehicle inside the previous region (re-entrance on the original region must also be signaled by the “Transition detection” block).

The definition of this architecture, which seems much more appropriate for autonomous mapping of regions with the characteristics of the maerl fields found at the Orkney Islands, is not yet fully accomplished. Its development has been studied using simulated vehicle motions and the video sequences recorded at Orkney in August 2000, during the first sea trial of the SUMARE project. The method is formally described in the PhD thesis [rolfes2003] and preliminary results have been presented on several papers published by members of the project [RolfesRendas2002, RolfesRendas2002a], and the interested

reader is directed to these references. Hopefully, preliminary results of this approach will be presented on the project's final report on the maerl mapping application (deliverable D8.2.)

Contour detection

In this Chapter we present the signal processing sub-system used for tracking contour lines corresponding to abrupt transitions from one kind of benthic habitat to another.

As it is shown in Figure 4, this sub-system is mainly composed of two blocks: one that performs unsupervised segmentation of consecutive video frames, and another that chooses, amongst the contours found in the present frame, the contour that the vehicle is following. The choice of the contour that is tracked by the vehicle makes use of an estimate of the vehicle velocity, which can be obtained either from integrating inertial measures using the vehicle dynamic model, or from developing a mosaic of the video frames acquired so far. The present implementation uses the second approach.

The Chapter is organized in the following way. We first recall the basic principles of the unsupervised video segmentation technique, which is fully described in Deliverable D3.3a of the project. The subsequent section discusses mosaicing of the segmented images, from which a course estimate of the vehicle displacement on the horizontal plane is obtained. We stress that this is a course mosaicing technique, whose goal is to produce, on-line, an updated estimate of the actual vehicle direction of motion. It is not an alternative to the work on video segmentation already presented in another deliverable of the project (D3.1), whose goal is to produce high quality video mosaics of the sea floor, suitable for human visualization. Finally, the last section of the Chapter presents the criterion used to determine which image contour is tracked by the vehicle at each time instant. The interaction between these blocks is shown in Figure 6 below.

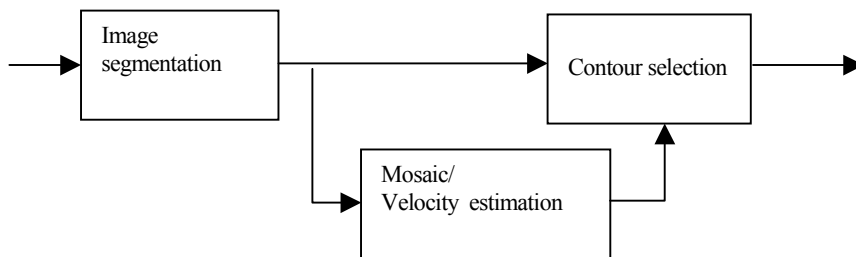


Figure 5 : selection of contour to track.

Video segmentation

We briefly remember here the basic principles behind the video segmentation technique, referring the interested reader to deliverable D3.3.a for a full presentation of the algorithm.

The goal of unsupervised video segmentation is to partition each frame of a video sequence into a small number of *homogeneous* regions. We consider that a region is homogenous if the distribution of gray level values in each small neighborhood of each pixel is almost constant.

Let I_{ij} be the gray level of the pixel (i, j) of an image I . Let K be the number of distinct classes in the image, and $\mu_k, k=1, \dots, K$ be the probability distribution of the pixels intensities associated to each class. Denote by μ_{ij} the histogram of gray level values of the pixels of image I inside a neighborhood W centered at pixel (i, j) . Then according to our assumption,

$$D(\mu_{ij}|\mu_k) \approx 0 \Leftrightarrow (i, j) \in \mathfrak{R}_k,$$

where \mathfrak{R}_k denotes the set of pixels of image I associated to class k , and $D(v|\mu)$ is a suitable similarity measure in the space of probability laws.

Based on this definition, we used formal tools of Statistical Signal Processing to determine the optimality criterion $C(\mathfrak{R}_k, k=1, \dots, K)$ that the optimal partition $\{\mathfrak{R}_k\}$ must verify: the weighted sum of the entropy of the probability distribution associated to the classes must be minimal:

$$\{\mathfrak{R}_k\} = \underset{\{\mathfrak{R}_k\}}{\operatorname{argmin}} C(\mathfrak{R}_k, k=1, \dots, K), \quad C(\mathfrak{R}_k, k=1, \dots, K) = \sum_{k=1}^K \alpha_k H(\hat{\mu}_k),$$

where α_k is the percentage of pixels associated to class k , $\hat{\mu}_k$ is the estimate of the probability law characteristics of class k ,

$$\hat{\mu}_k = \frac{1}{|\mathfrak{R}_k|} \sum_{(i, j) \in \mathfrak{R}_k} \mu_{ij},$$

and $H(\mu_k)$ is the Shannon entropy of the probability law associated to class k , defined by

$$H(\mu_k) = \sum_{i=1}^L \mu_k(i) \ln[\mu_k(i)],$$

where L is the number of gray levels in the image (usually 256 for black and white images). Note that the criterion $C(\mathfrak{R}_k, k=1, \dots, K)$ depends only indirectly, through the estimated classes' probability laws, in the classification of each pixel. The formal probabilistic framework adopted for this study revealed also the precise meaning of the "measure of similarity between probability laws" $D(\cdot, \cdot)$: it must be understood as the Kullback-Leibler divergence between distributions, defined by

$$D(v|\mu) = \sum_{i=1}^L v_i \ln\left(\frac{v_i}{\mu_i}\right).$$

Global optimisation of this criterion leads to a combinatorial optimisation problem, which cannot be solved in real-time. However, we were able to show that an iterative algorithm of manageable complexity can be proved to converge to a local minimum of $C(\mathfrak{R}_k, k=1, \dots, K)$, and to define an efficient initialisation procedure that minimises

problems of convergence to local minima. This work is reported in detail in deliverable D3.3.

Alone, the criterion $C(\mathfrak{R}_k, k=1, \dots, K)$ above does not enable determination of the number of classes (K) into which the image should be partitioned. In deliverable D3.3 we study several distinct approaches to this difficult problem: (i) based on a test on the distribution of the distance (actually the Kullback-Leibler divergence) between the local probability distributions μ_{ij} of the regions associated to the class and its representatives $\hat{\mu}_k$, (ii) the use of the Minimum Description Length Criterion of Rissanen, which considers the addition of a term penalizing complex models, and (iii) the use of prior stochastic models on the regions geometry (Connected Component Fields). These three approaches lead to algorithms of different computational complexity, and which one should be preferred depends on the amount of prior knowledge about the sea-bed characteristics. In this report we use the first one, which exploits the fact that the set of distances

$$d_{ij} = D(\mu_{ij} | \hat{\mu}_k), \forall (i, j) \in \mathfrak{R}_k$$

should follow an exponential distribution. Our algorithm (see Deliverable 3.3.a) adaptively changes the number of classes according to a statistical test on these distances.

We show below an example of the segmentation of an image taken at Orkney (maerl beds) with a multi-resolution version of this algorithm.

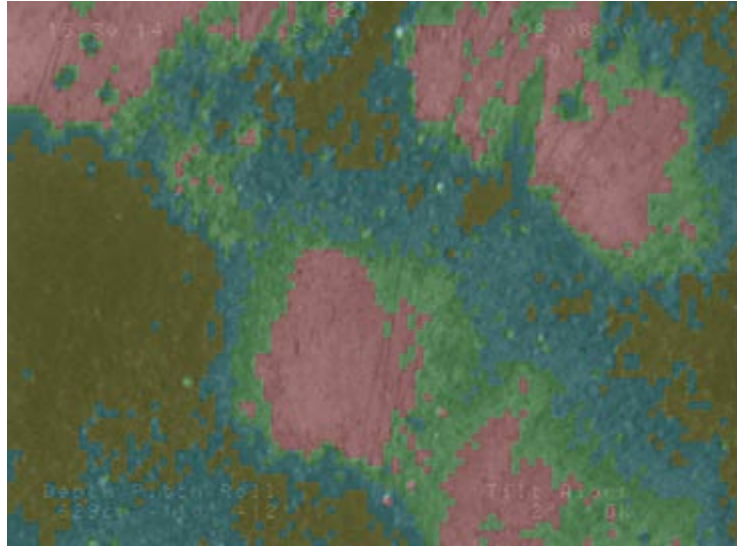


Figure 6: unsupervised segmentation of an image of a maerl field.

While being able to automatically adjust the number of classes to the complexity of each image is important to accommodate variations of the natural regions being tracked, and entrance of other regions into the camera's field of view, abrupt changes in the set of labels into which the image is partitioned may produce an unstable behaviour of the

contour tracking algorithm. We will see below how we handled this problem, by using a robust metric for the association of pairs of consecutive partitioned images.

Mosaicing

This section presents the algorithm that incrementally builds a mosaic of the segmented images, and from which we infer the *displacement of the robot in the horizontal plane* (remember that no global position information is available in the Phantom platform, and thus the perceptual sensors are the only way to assess the horizontal motion of the vehicle).

Our mosaicing algorithm integrates all sensor data available, in particular:

- the vehicle compass, which gives us the orientation of the camera with respect to the (magnetic) North;
- the vehicle's altimeter, which enables us to define a metric on the image plane.

Neglecting the robot roll and pitch, as well as altitude variations between consecutive frames, the rigid motion of the robot between two consecutive instants (measured in pixels in the image plane) maps pixel (i, j) of image I_n into pixel (i_{n+1}, j_{n+1}) given by

$$\begin{pmatrix} i_{n+1} \\ j_{n+1} \end{pmatrix} = R(\Delta\theta) \begin{pmatrix} i_n \\ j_n \end{pmatrix} + \begin{pmatrix} u_x \\ u_y \end{pmatrix}$$

where $\Delta\theta$ is the (measured) rotation of the robot between the two frames, and $u = (u_x, u_y)$ the displacement observed in the image plane. Let $R(\Delta\theta, u)$ denote the set of pixels (i, j) of image I_n that are mapped inside the image I_{n+1} by the above equation:

$$R(\Delta\theta, u) = \left\{ (i, j) \in I_n : R(\Delta\theta) \begin{pmatrix} i \\ j \end{pmatrix} + u \in I_{n+1} \right\}.$$

Using two consecutive segmented images, which are a field of labels

$$L_{ij}^n \in \{1, \dots, K_n\}, L_{ij}^{n+1} \in \{1, \dots, K_{n+1}\}$$

L_{ij}^n associating pixel (i, j) in image I_n with class k with probability distribution μ_k^n , we estimate the *displacement in the image plane* by searching for the vector u_n that minimizes the following criterion

$$u_n = \underset{u}{\operatorname{argmin}} \sum_{i, j \in R(\Delta\theta, u)} D(\mu_{L_{ij}^n} | \mu_{L_{R(\Delta\theta)(ij)+u}^{n+1}})$$

i.e., of the sum of the Kullback distances between the probability distributions associated to the pixels that are put in correspondence by vector u . Note that the computational complexity of this optimization problem is simplified by the fact that, for all possible value of u , the criterion depends only on the $K_n \times K_{n+1}$ values of the Kullback divergences between the classes detected in each image.

We show below a video mosaic (actual observed length is about 30 meters) built in the Orkney Islands during the sea trials done by the project in September 2002.



Figure 7 mosaic of segmented images (Orkney Islands, 2002)

The mosaic of the first 50 frames of the actual image (obtained by simply overlapping successive image frames using the displacement inferred from the correspondence of the segmented images) is shown below

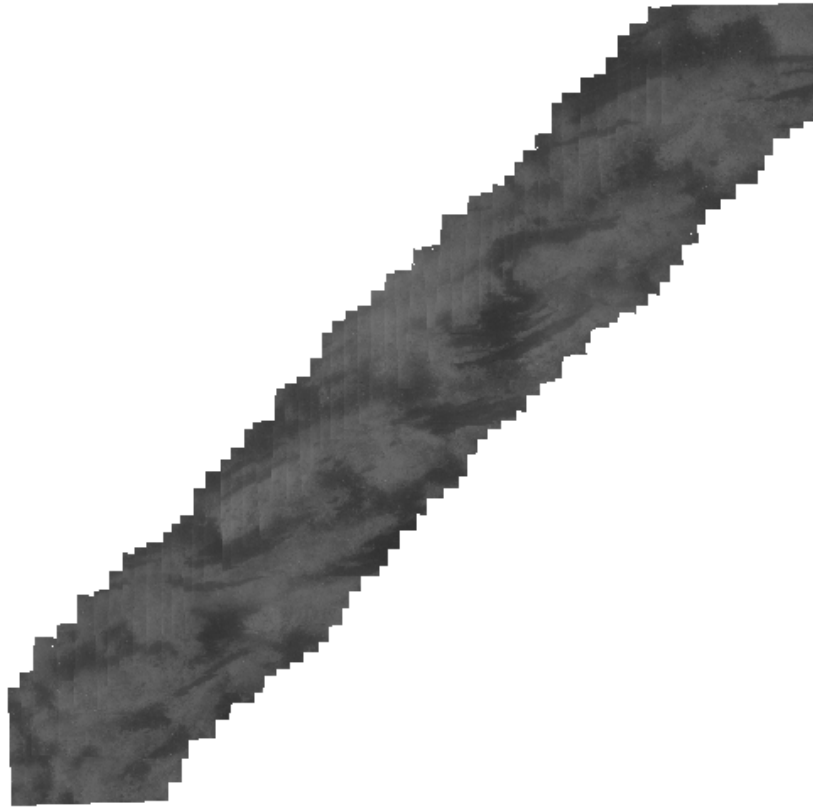


Figure 8 : mosaic of the 1st 50 frames of the images of the sequence on Figure 7.

The same technique has been used to produce video mosaics of the contours between posidonia and sand at Villefrance-sur-mer (France). We present below two examples, the first using the segmented images, and the other showing the result on the actual video frames.

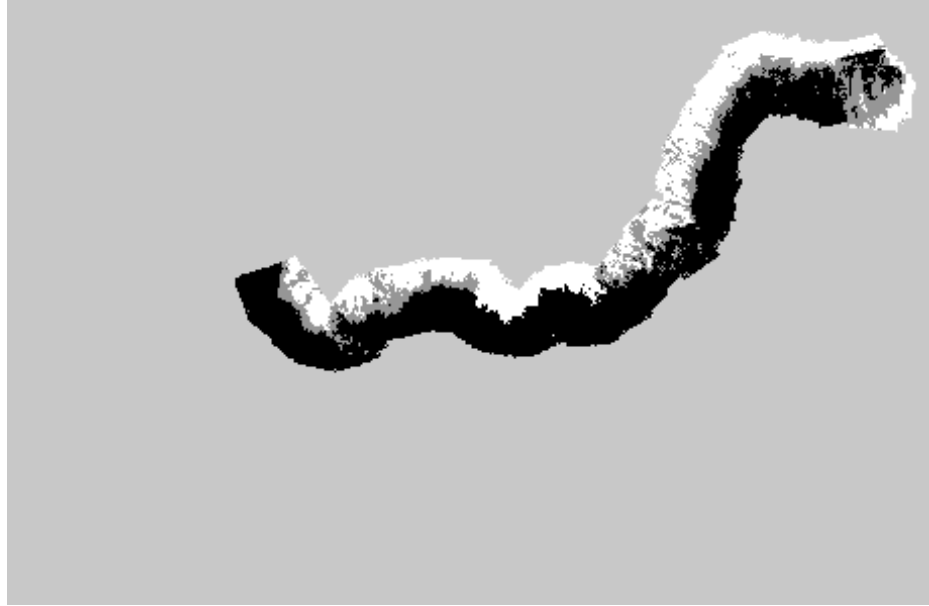


Figure 9 : mosaics of contours between posidonia and sand at Villefranche-sur-mer, France.

The estimated vehicle displacement measured in the image plane, u , is then transformed, using knowledge of the camera intrinsic parameters and the altimeter data, and subsequently fused, with other sensor data, to update the global position and orientation estimates of the vehicle.

The estimates u_n are also used, as we see in the next section, to determine which of the contours of the regions found in image I_{n+1} is the one being tracked.

Contour selection

The segmentation of each video frame partitions the current image in a series of regions. To guarantee continuity of the tracking task, we must choose, amongst the contours of all these regions, which one the vehicle is tracking.

Although some work has been done to automate the decision on whether there is a sufficiently good contour to track, and select the best one, the present implementation that has been used in field trials relies on the human user to specify (by mouse clicking on the segmented image) which contour he/she wishes to track. We can assume thus, that an initial contour C^0 , represented as a chain of pixel coordinates in the image, is chosen by the user in the first frame.

Let $\{C_i^n\}_{i=1}^{M_n}$ be the set of open contours (we eliminate closed contours in the image in a first step) in image I^n whose extremes fall in the image edges (outgoing contours), and u_n the estimated displacement (in pixels) of I_{n+1} with respect to I_n as defined in the previous section. Denote by $\Delta\theta$ the variation of the orientation of the camera (which is solidary with the robot) between the two images. Since, according to our criterion for estimation of u , C^0 should have been translated by the same vector u , we select the next contour to track by searching for the one that is closest to $R(\Delta\phi)C^{n-1}+u$:

$$C^n = \operatorname{argmin}_{i=1, \dots, M_n} \sum_{j \in R\Delta\theta(u)} \min_j \left(\|C_i^n(j) - R(\Delta\phi)C^{n-1}(j) - u\|^2 \right).$$

Note that the sum is done over all points of the previous contour that should fall inside the current frame. To each predicted position, we associate the closest point of the contours found in the present frame. In this way, small spurious contours that may fall near the predicted position of the contour being tracked, are penalized.

The next figure illustrates the determination of the criterion (using one of the windows of the real-time application). The window displays the current image in the left, and two smaller areas display the last two segmented images: the top one is the segmentation of the current image (shown in the left), and the lower one the previous segmented image. In the example below, three distinct classes have been identified by the segmentation algorithm coded by the green, red and black regions in the smaller images.

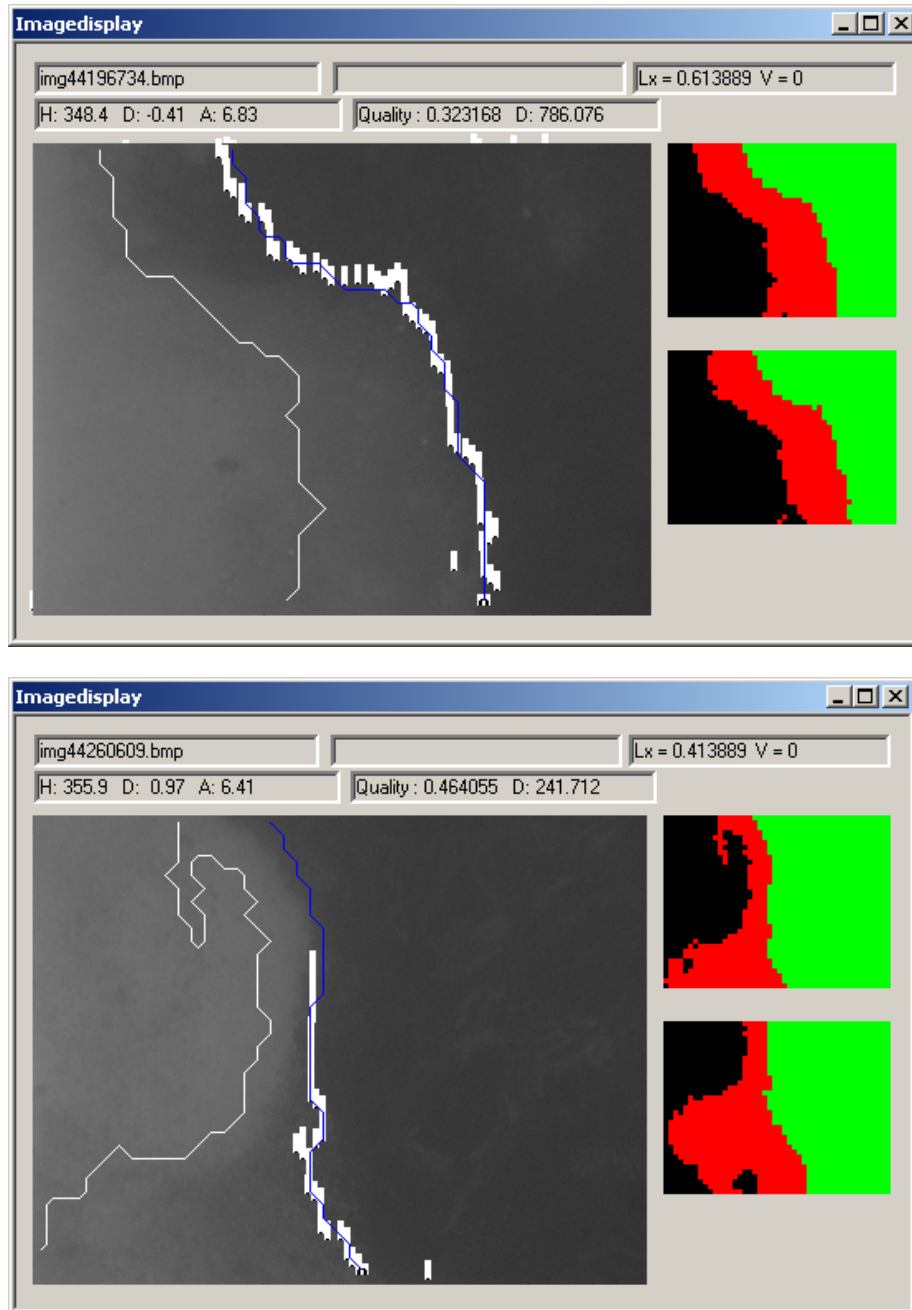


Figure 10: frame-to-frame association of the tracked contour.

The lines superposed to the acquired image are the open contours of the detected regions which cross the image boundary. The blue line is the contour chosen for tracking. The white squares represent the prediction of the position in the current image of the points of the contour tracked in the previous frame. While in the top figure the displacement of the robot is very small, resulting in a predicted contour close to one of the detected contours, in the second case the robot motion between the two frames is considerable, explaining why the prediction of the contour is incomplete. The next chapter describes how the selected contour is used to generate the appropriate control signals.

Visual Controller

In this Chapter we present the architecture of the control system block in Figure 4, whose goal is to keep the origin of the camera over the contour being tracked.

The body motion of the Phantom 500 is controlled with the usual decoupled structure, based on three basic separate control loops :

- *altitude* controller (loop closed around the altimeter measure),
- *heading / rate* controller (loops closed around the compass and the gyroscope, respectively) ,
- *surge* open loop controller (open loop, since we have no speed sensor, such as a Doppler-log).

This control structure is appropriate for the Phantom 500. This vehicle is equipped only with two horizontal thrusters and a vertical thruster : the number of actuators is smaller than the number of degrees of freedom. Pitch, roll and sway dynamics are not controlled but assumed to be intrinsically stable. The three thrusters are speed controlled. The 3 decoupled design models for the body motion are reasonably simple and capture the main features of the vehicle dynamics.

The basic control structure (inner body motion control loops and the feed forward surge controller) in loop with the plant is depicted in Figure12. The input signals u_r , ψ_r , h_r represent the surge reference, the heading reference and the altitude reference, respectively. For a reasonable set of dynamics, we assume that the measured signals ψ , h

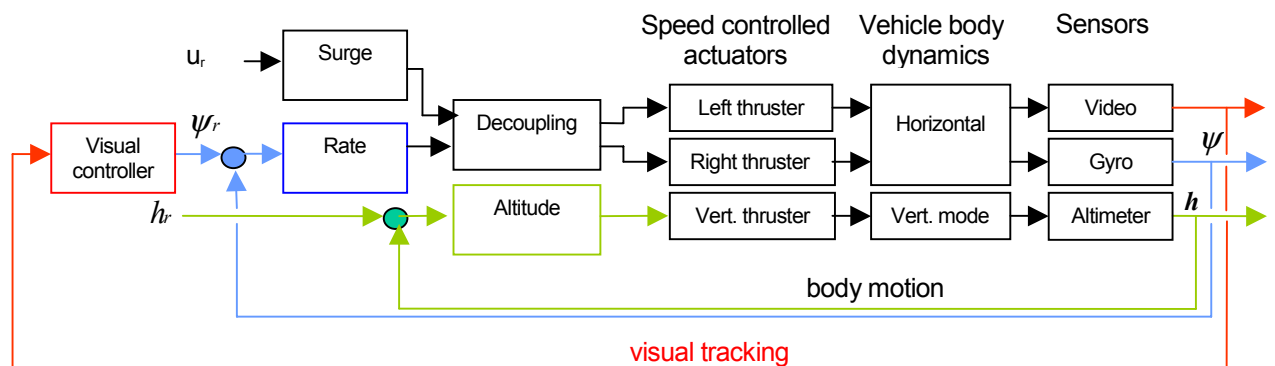


Figure 11: complete control structure

track the reference signals ψ_r, h_r .

The contour following task is decomposed into the following control objectives:

- i. drive the vehicle at constant altitude from the contour
- ii. keep the camera origin above the contour line.

When the basic control loops are active, the control objective (i) is ensured by imposing a fixed altitude reference signal h_{ref} (1 m, for instance). The control problem reduces to steering the vehicle in the horizontal plane. The approach retained consists in fixing a constant surge for the vehicle and to minimize a conveniently defined (see below) distance in the image. The trajectory of the vehicle in the global frame and the sea-bed contour are represented in Figure 13.

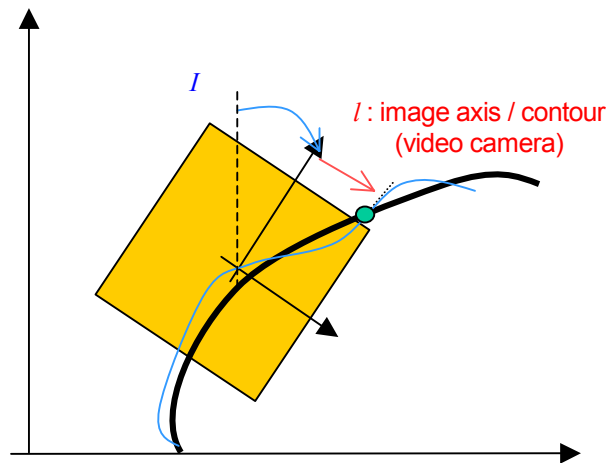


Figure 12: control strategy

The proposed approach to impose the desired control objective consists in minimizing the distance l between the contour and the axis u of the camera. This distance can be interpreted as the error of the visual controller. A physical interpretation of the control strategy is to steer on the left the vehicle when signal l is positive. Naturally, the control signal for this loop is the vehicle yaw *rate*.

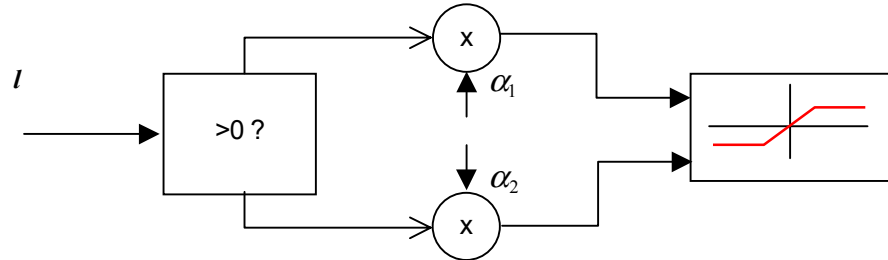


Figure 13: visual tracker.

The visual controller is a simple asymmetric gain followed by a saturation, as shown in Figure 14. The asymmetric gain compensates the fact that the camera origin is not aligned with the vehicle's axis of symmetry, inducing an asymmetric behaviour when turning left and right.

Experiments at sea.

We present in this section results of contour tracking experiments performed in Villefranche-sur-mer, in the South of France, using the ROV Phantom.

In Figure 15 we show the video mosaic built **during** one of the contour tracking experiences, where the tracked contour is the separation between the sandy bottom and Posidonia. In this Figure, the vertical axis is aligned with the North, and the horizontal axis points East. Tracking is aborted at the end of the experiment, since there is no clear contour present in the last frames (top right extreme of the figure).



Figure 14: contour between Posidonia and sand (Villefranche-sur-mer)

The ROV starts its observation from the South-East extreme, tracing a curved path around one patch of Posidonia (Posidonia is maintained in the right side of the image). Figure 16 shows the evolution of the error signal ℓ and of the reference rate input to the low-level rate controller (running in the dSpace board).

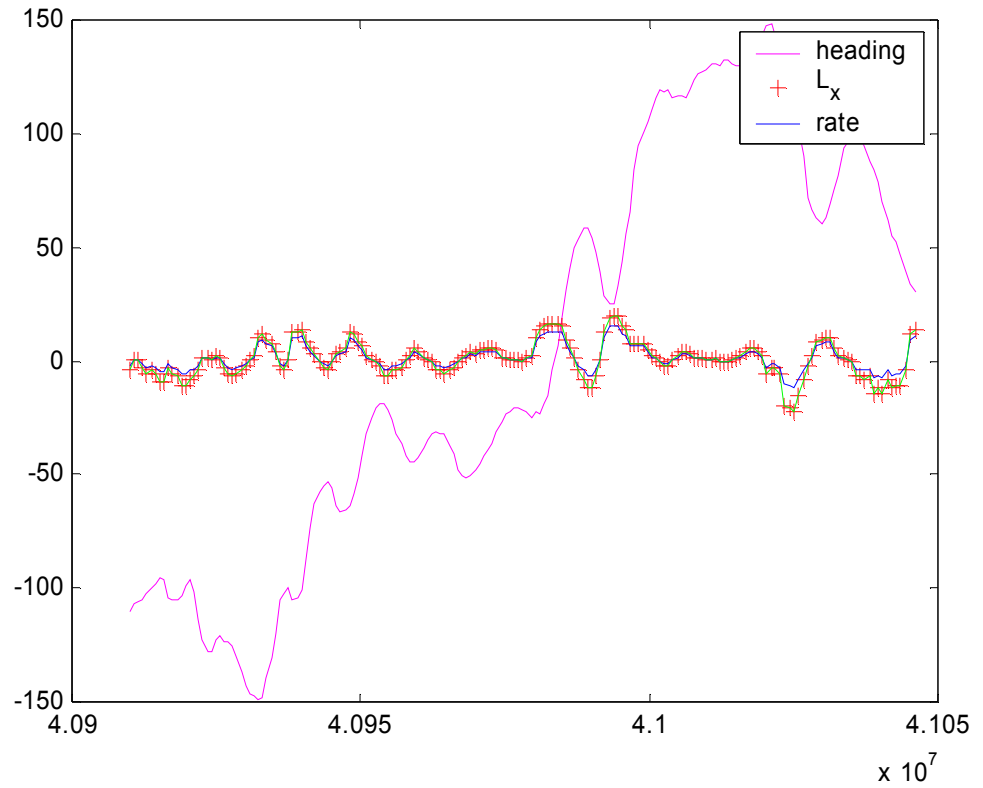


Figure 15: evolution of error signal and rate reference.

The next Figure shows, superimposed, the reference signal and the actual (measured) rate indicated by the robot's gyroscope (top) and the commands sent to the left and right motors of the Phantom (rotation rates). As it can be seen, the system is never under saturation, due to the choice of controller gains across the entire processing chain.

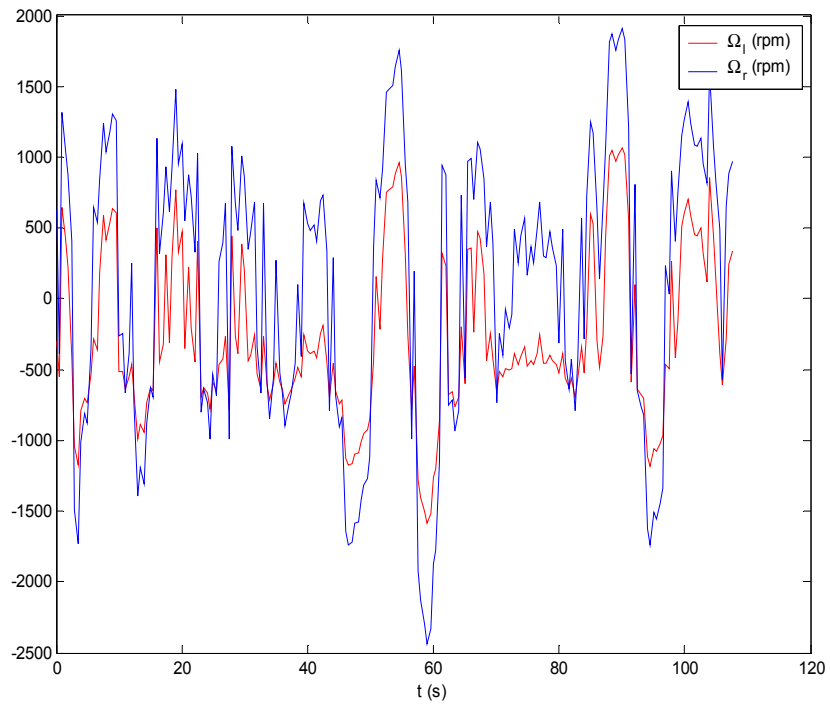
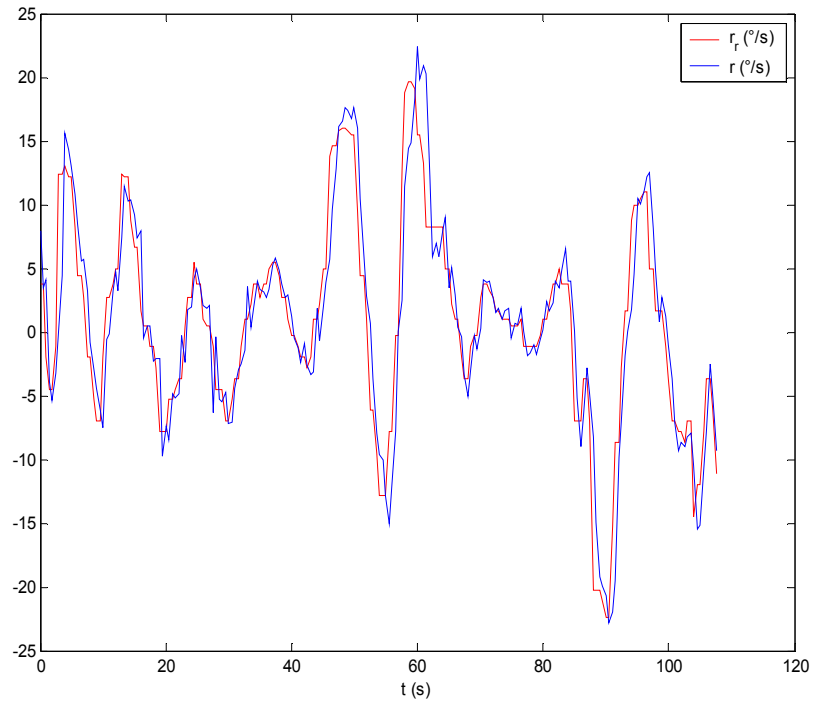


Figure 16: reference rate and mesured signal (top) and motor commands (bottom).

The next Figures show the corresponding plots for another tracking experiment in the same region. Note that even for this relatively straight contour line, the very local field of view of the camera (at an altitude of 3 meters for this experiment), coupled with the local curvature of the contour, induce an oscillating behaviour when tracking.

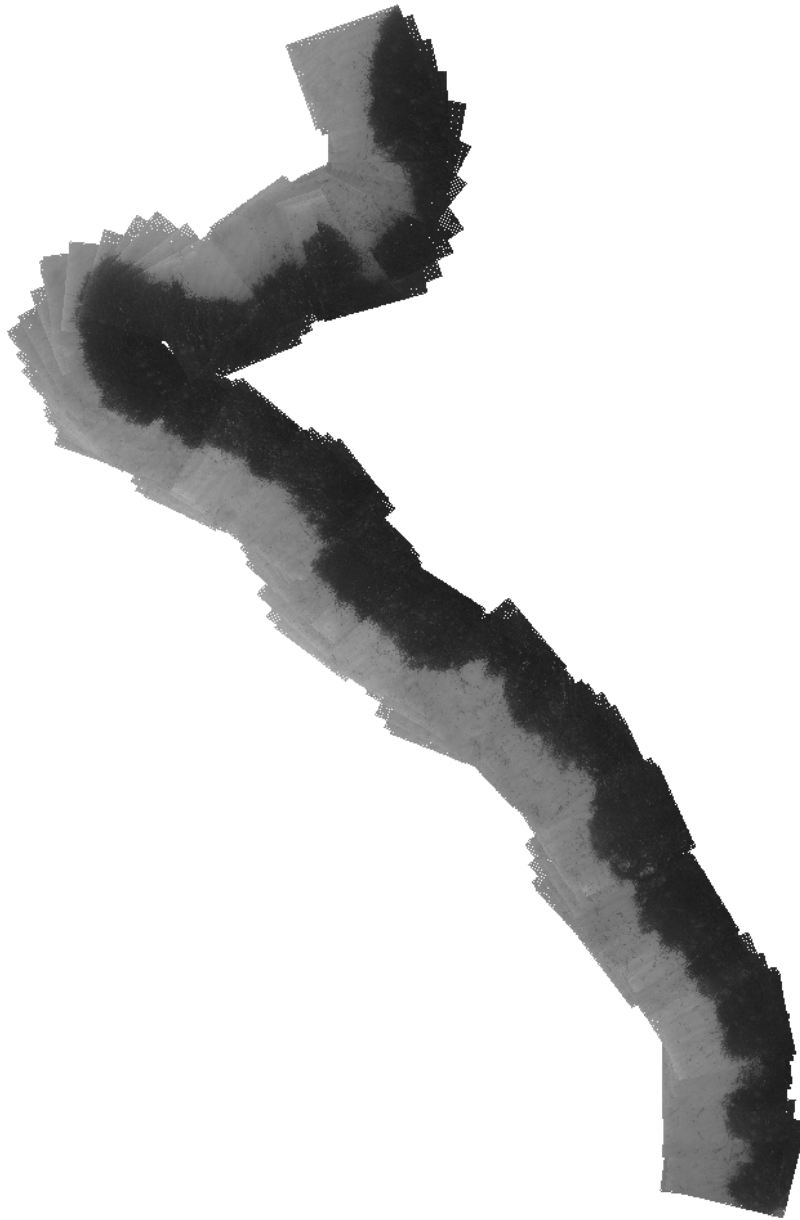


Figure 17: contour between Posidonia and sand (Villefranche-sur-mer).

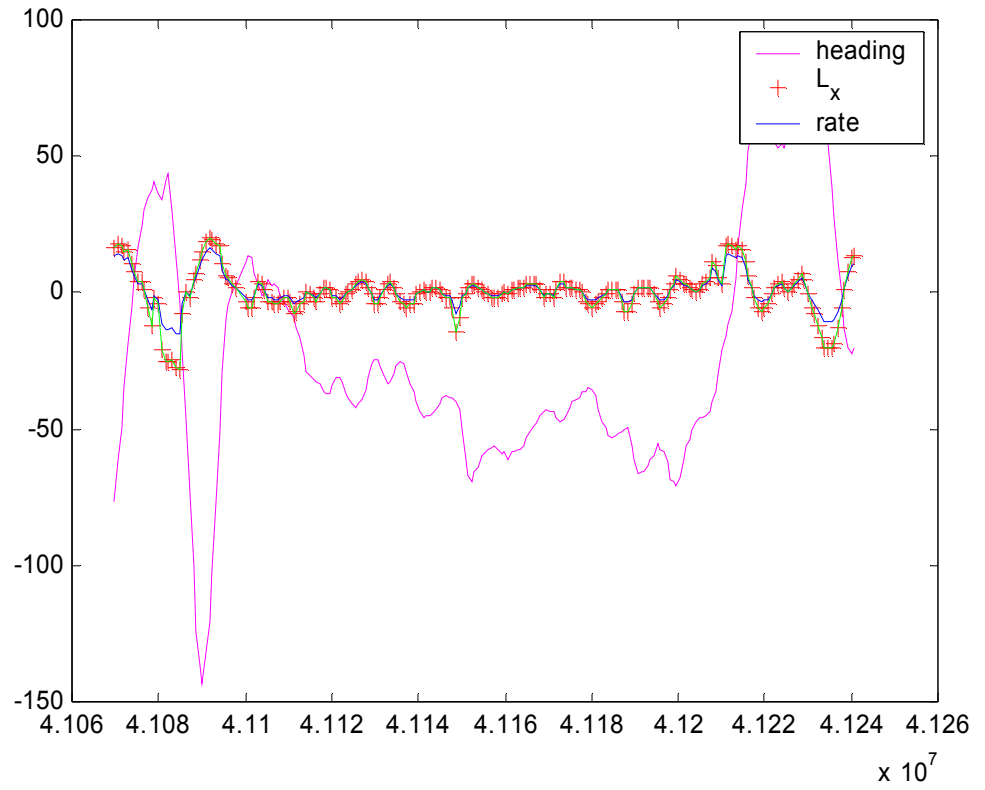


Figure 18: evolution of error and control signals.

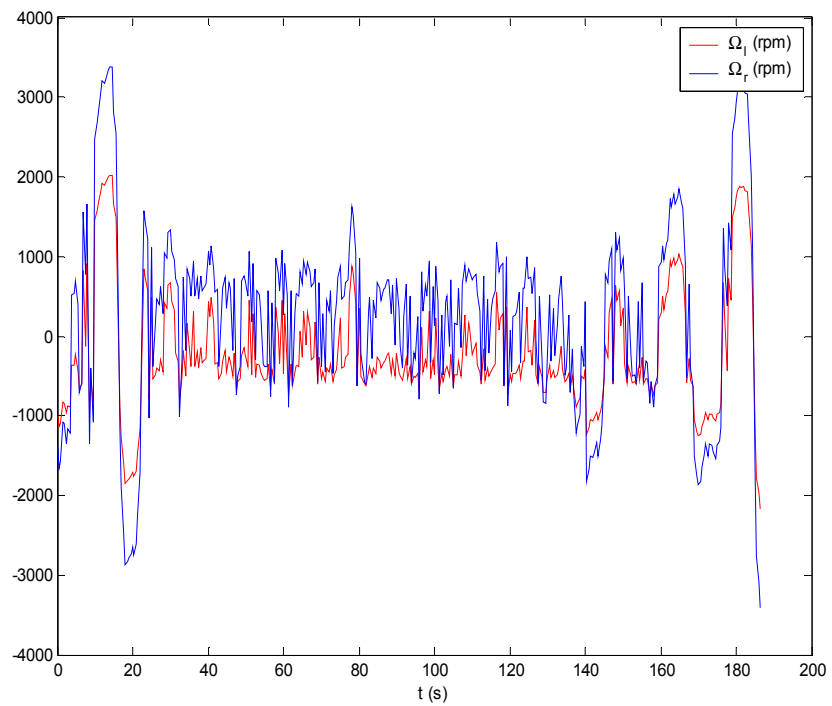
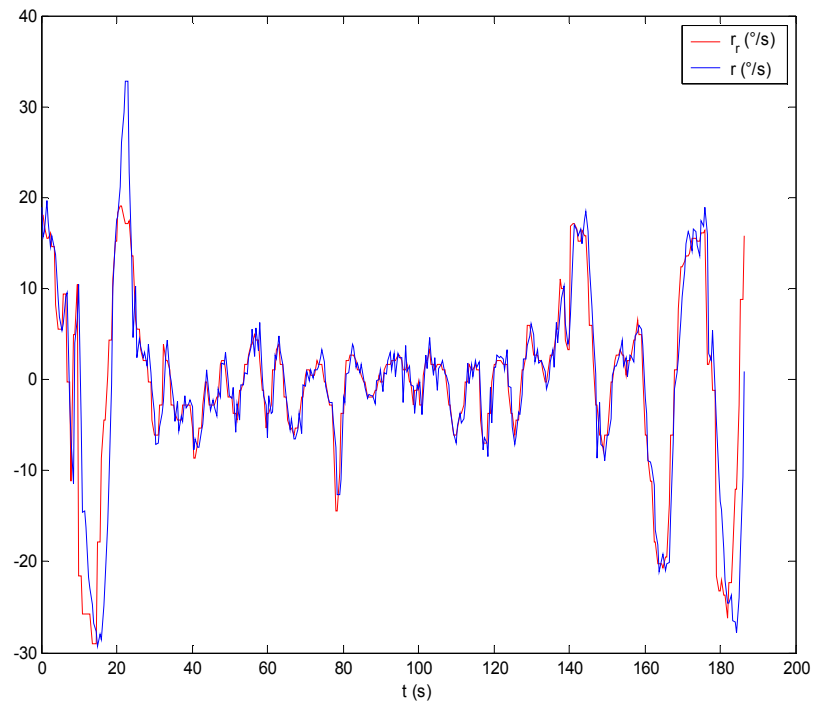


Figure 19: Reference rate and measured signal (top) and motor commands (bottom).

Finally, we present in the next two Figures a third experiment conducted at an higher altitude (3 meters), showing the resulting mosaic and the command signal and the robot's heading.



Figure 20: acquired mosaic.

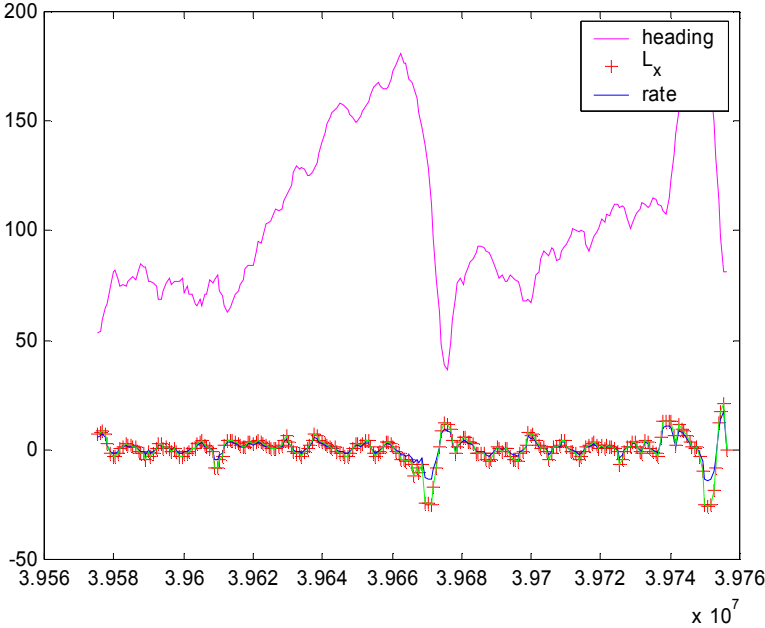


Figure 21: control signals.

Contour Tracking with Altimeter

In this Chapter we present the work that has been done on the definition of algorithms for tracking iso-depth lines, using the measures acquired by an altimeter.

Problem statement

The goal of this sensor driven mode is to guide an autonomous platform equipped of an altimeter sensor along lines of constant sea depth. Similarly to what we did to define the visual tracker in the preceding chapters, our implementation of the altitude tracker imposes that the vehicle be kept at constant depth (below sea level) during the observation of the contour. This choice also minimizes pitch variations which are directly mapped into noise in the altitude measures. Thus, we assume in this section, in a simplistic manner, that, for all time instants t_k ,

$$z(t_k) = D_{ref}$$

where D_{ref} is the desired vehicle depth.

Note: Keeping the vehicle at constant depth has the advantage, for the Phantom, of minimizing the excitation of uncontrolled pitch dynamics. For Mauve, it is mandatory, since depth control imposes pitch variations, inducing large uncertainties with respect to the point of the sea bottom being sampled. The value D_{ref} should be chosen such that, at the depth of the desired contour line, the error induced by residual pitch variations be acceptable.

Let $h(x,y)$ denote the bathymetry (sea depth) surface at position x,y (in a Cartesian global frame), and let x_k, y_k denote the position of the robot (that we assume for the moment to be coincident with the sensor reference frame) at instant t_k , see Figure 23.

Denote by θ_k, φ_k and ϕ_k , respectively, the vehicle's yaw, pitch and roll attitude angles, and by x_k, y_k , and z_k the vehicle's coordinates in the inertial frame. The altimeter measures, in the body frame, the distance d_k of the vehicle to the sea bottom. To evaluate the sea bottom depth $h(x_{in_k}, y_{in_k})$ at each position x_{in_k}, y_{in_k} in the inertial frame, one needs to convert the altimeter measurements, and the measured position from the body to the inertial frame:

$$\begin{bmatrix} x_{in_k} \\ y_{in_k} \\ h(x_{in_k}, y_{in_k}) \end{bmatrix} = R_{yaw}^T R_{pitch}^T R_{roll}^T \begin{bmatrix} 0 \\ 0 \\ d_k \end{bmatrix} + \begin{bmatrix} x_k \\ y_k \\ z_k \end{bmatrix}$$

where

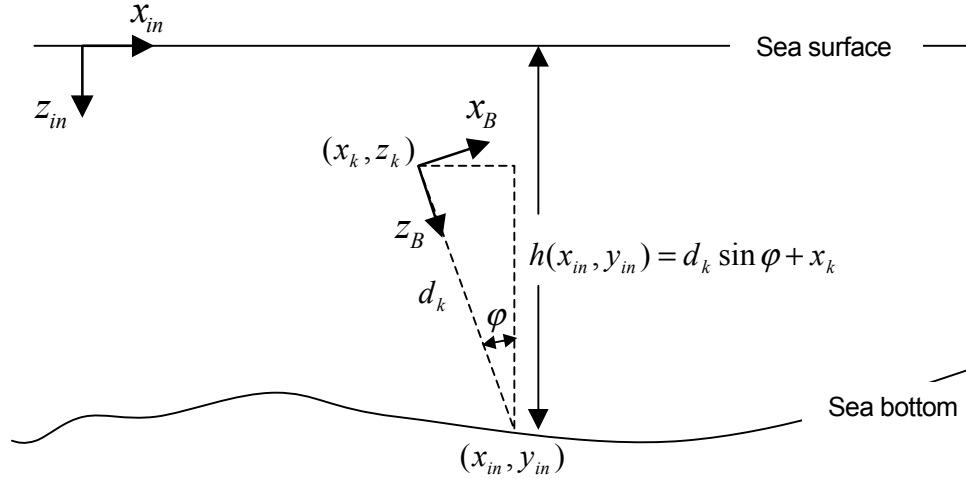


Figure 22: Problem geometry.

$$R_{yaw} = \begin{bmatrix} \cos \theta_k & \sin \theta_k & 0 \\ -\sin \theta_k & \cos \theta_k & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad R_{pitch} = \begin{bmatrix} \cos \phi_k & 0 & \sin \phi_k \\ 0 & 1 & 0 \\ -\sin \phi_k & 0 & \cos \phi_k \end{bmatrix},$$

$$R_{roll} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi_k & \sin \phi_k \\ 0 & -\sin \phi_k & \cos \phi_k \end{bmatrix},$$

that is

$$\begin{bmatrix} x_{in_k} \\ y_{in_k} \\ h(x_{in_k}, y_{in_k}) \end{bmatrix} = \begin{bmatrix} -\cos \theta_k \sin \phi_k \cos \phi_k + \sin \theta_k \sin \phi_k \\ -\sin \theta_k \sin \phi_k \cos \phi_k - \cos \theta_k \sin \phi_k \\ \cos \phi_k \cos \phi_k \end{bmatrix} d_k + \begin{bmatrix} x_k \\ y_k \\ z_k \end{bmatrix}$$

Remark that for zero roll and pitch attitude angles, that is when the vehicle is parallel to the sea surface, the (x, y) coordinates of the point for which the sea floor depth is measured coincides with the vehicle position and the bathymetry measurement is just the sum of the vehicle's depth with the altimeter measurement.

In the sequel, we assume that the roll and pitch angles are close to zero, such that

$$h(x_{in_k}, y_{in_k}) \cong d_k + z_k.$$

The control problem can thus be formulated as achieving the simultaneous control objectives:

- i. **keep** $z_k = z_{ref}$

ii. **keep** $d_k = h_{ref} - z_{ref}$

We present in the next section the control architecture that achieves these two goals.

Approach

We represent in Figure 24 the architecture of the control system implemented on-board the ROV Phantom 500, whose goal is to guide the vehicle along iso-depth lines using altitude measures.

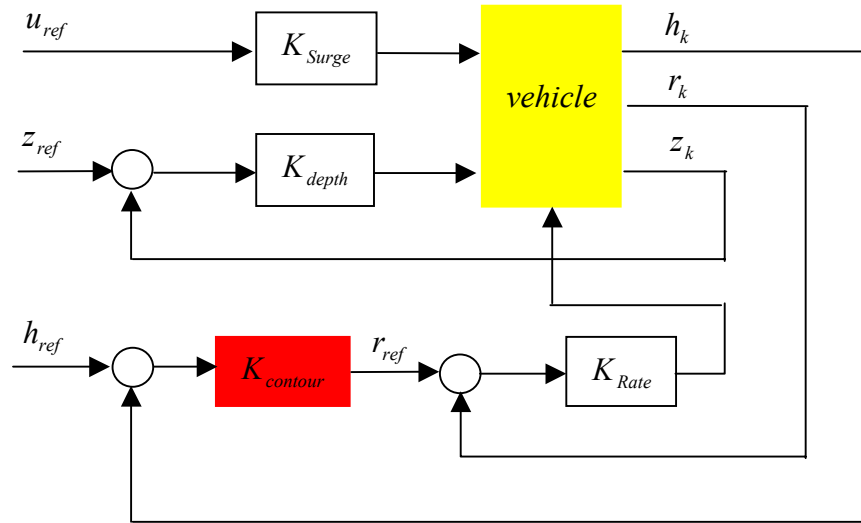


Figure 24: Control architecture for iso-depth lines tracking using altimeter measurements

The body motion of the vehicle is controlled using three decoupled control loops: closed-loop depth and rate controllers and open-loop surge controller. When the basic control loops are active, and after detecting the iso-depth bathymetric line to be tracked, which is characterized by a pre-specified altitude h_{ref} , the closed-loop contour tracker (block in red in Figure 24) is responsible for generating the rate reference signal to be imposed to the rate control system. For constant linear velocity, u , this corresponds to controlling the trajectory curvature κ :

$$\kappa = r_k / u$$

To define the contour tracking mission using altitude measures, the following parameters need to be specified:

1. u_{ref} -- vehicle's linear velocity;
2. z_{ref} -- vehicle's depth;
3. h_{ref} -- altitude of the iso-depth bathymetric line to be tracked;
4. $LeftRight$ -- indicates if the vehicle turns left ($LeftRight = -1$) or right ($LeftRight = +1$) once the iso-depth line to be tracked is detected.

The first two parameters characterize the body motion and are required for any type of mission, while the two last ones are specific to the contour controller.

Contour tracker

Let us assume that the contour line to be tracked is located at a surface locally planar with a non-zero slope. This means that the error $h_{ref} - h_k$ has opposite signs at each side of the contour.

The contour controller is characterized by a proportional-derivative control law:

$$r_{ref_k} = [K_P(h_{ref} - h_k) - K_D(h_k - h_{k-1})]S$$

where $S = LeftRight Var_h$. In the previous expression, *LeftRight* stands for the input parameter that defines the direction of turning when the reference altitude is detected, and Var_h indicates if the vehicle motion previous to contour tracking is being done following increasing ($h_{ref} - h_k > 0 \rightarrow Var_h = -1$) or decreasing ($h_{ref} - h_k < 0 \rightarrow Var_h = +1$) altitudes. Remark that variable S is positive (+1) or negative (-1) when the contour is followed keeping the highest or the lowest regions, respectively, on the right or left side of the vehicle. This allows us to change the sign of r_{ref_k} in order to conduct the vehicle towards the contour line.

Before proceeding with the design of the controller gains K_P and K_D , and to get some insight into the behaviour of the adopted control law, let us observe some simulation results obtained for the planar terrain with non-zero slope represented in Figure 25. In Figure 26 we plot the vehicle's trajectory when tracking the line of $h_{ref} = 0$ in the planar surface of Figure 25, for two different values of (K_P, K_D) . Remark that, although both trajectories exhibit a stable behaviour, the frequency of oscillation, maximum amplitude and damping are quite different. This leads to the need of defining a strategy to design the control system gains: proportional gain, K_P , and differential gain, K_D .

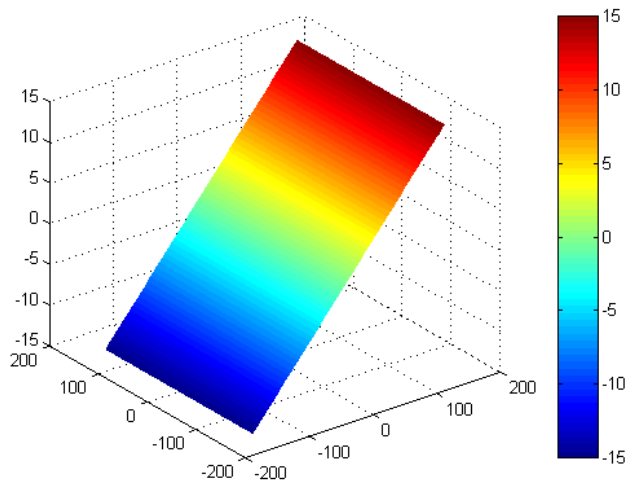


Figure 25 : planar environment.

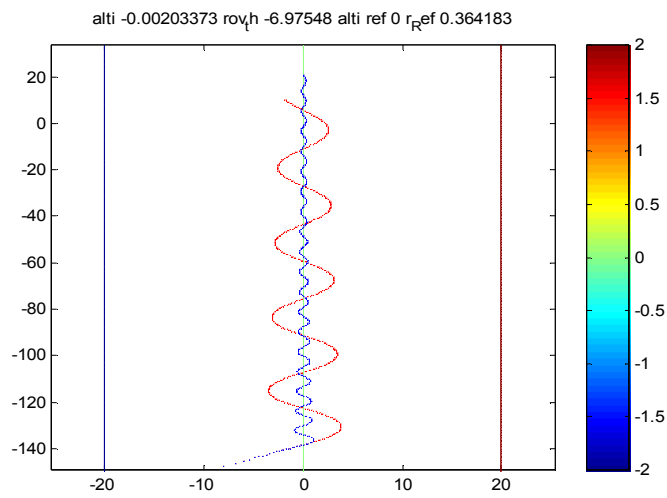


Figure 26: Vehicle trajectory when tracking line of zero altitude in the planar surface:
 $K_p = 50$, $K_D = 1000$ - blue line, and $K_p = 3.3$, $K_D = 95$ - red line

To characterize the vehicle's trajectory in the (x, y) plane, let us consider the simplest case of the planar surface where the iso-depth lines are parallel to the y coordinate. In this case the bathymetry can be modeled by

$$h(x, y) = \rho y + h_{ref}$$

where ρ represents the slope of the terrain. Also, let us assume that the vehicle trajectory $y(x)$ can be approximately modeled by a damped sinusoid:

$$y(x) = A \sin(fx) e^{-\alpha x}$$

Note that the trajectory in the (x, y) plane can, in continuous time, be modeled by

$$\frac{dx_\ell}{d\ell} = \cos \theta_\ell, \quad \frac{dy_\ell}{d\ell} = \sin \theta_\ell \quad \text{and} \quad \frac{d\theta_\ell}{d\ell} = r_\ell$$

where, for constant linear velocity, u , we have $d\ell = u dt$. Computing the second derivative of

$$\frac{dy(x)}{dx} = \tan \theta$$

with respect to ℓ , and taking into account the previous relationships we conclude that

$$r_\ell = \frac{d\theta_\ell}{d\ell} = -(\alpha^2 + f^2) \cos^3 \theta_\ell y_\ell - 2\alpha \cos^2 \theta_\ell \frac{dy_\ell}{d\ell}.$$

The discrete time equivalent of the above equation, obtained taking into account that $r_i = r_\ell / u$, is

$$r_k = -u(\alpha^2 + f^2) \cos^3 \theta_k y_k - 2\alpha \cos^2 \theta_k \frac{y_k - y_{k-1}}{\Delta T}.$$

Finally, from the linear relationship between the sea floor altitude, h and the y coordinate of the trajectory, one gets

$$r_k = u(\alpha^2 + f^2) \cos^3 \theta_k \frac{1}{\rho} (h_{ref} - h_k) - 2\alpha \cos^2 \theta_k \frac{h_k - h_{k-1}}{\rho \Delta T}.$$

Comparing the previous equation with the proportional-derivative command law, we obtain

$$K_P = \frac{u(\alpha^2 + f^2)}{\rho} \cos^3 \theta_k \quad \text{and} \quad K_D = \frac{2\alpha}{\rho \Delta T} \cos^2 \theta_k$$

The previous gains, which are time-varying, depend on the time-evolution of the angle between the vehicle's trajectory and the line being tracked. A time-invariant, and thus linear, approach can be obtained by considering that θ_k is small $\forall k$, i.e., considering

$$K_P^{ap} = \frac{u(\alpha^2 + f^2)}{\rho} \quad \text{and} \quad K_D^{ap} = \frac{2\alpha}{\rho \Delta T}$$

Remark that, for $-\frac{\pi}{2} < \theta_k < \frac{\pi}{2}$, the linear (approximate) command r_k^{ap} is larger than r_k

leading to a trajectory that is always closer to the contour to be followed. This is clearly shown in Figure 27, where the trajectory in green is the obtained with the approximate time-invariant model, and the red one is the trajectory predicted by the damped sinusoid.

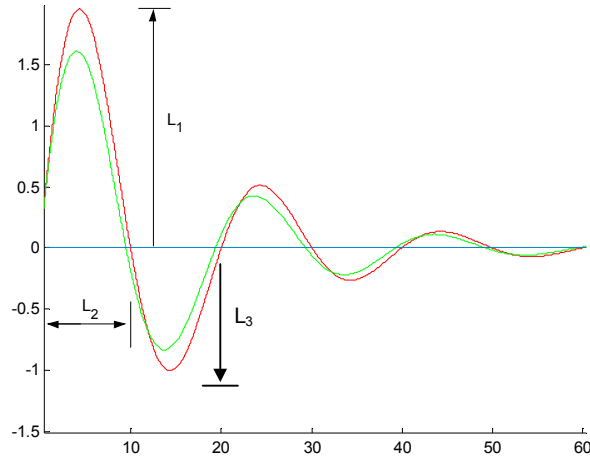


Figure 27: Approximate (linear model) trajectory in green , and trajectory given by the damped sinusoid.

The damped sinusoidal trajectory can be characterized by parameters L_1, L_2, L_3 represented in Figure 27. Taking into account the expression of $y(x)$, it is easy to show that

$$L_2 = \frac{\pi}{f} \quad \text{and} \quad \frac{L_3}{L_1} = e^{-\frac{\alpha\pi}{f}}$$

i.e.,

$$f = \frac{\pi}{L_2} \quad \text{and} \quad \alpha = \frac{1}{L_2} \ln\left(\frac{L_1}{L_3}\right)$$

The above relationships allow us to specify the frequency of oscillation of the trajectory around the contour (L_2 represents half of a period of the damped sinusoid) and the damping rate (which is characterized by the ratio L_3/L_1). However, the maximum deviation between the trajectory and the contour (L_1) cannot be fixed since it depends on the amplitude A of the damped sinusoid which is determined by the angle of arrival, θ_0 :

$$A = \frac{1}{f} \tan \theta_0 \quad \text{and} \quad L_1 = \frac{\tan \theta_0}{\sqrt{\alpha^2 + f^2}} e^{\frac{\alpha}{f} \arctan\left(\frac{f}{\alpha}\right)}.$$

Remark that θ_k represents, at each time k , the angle between the trajectory and the contour, which can not be *a priori* known by the robot.

In order to analyse the behaviour of the contour tracking system with respect to the design parameters L_1, L_2, L_3 , let us first consider the planar environment represented in

Figure 25. In Figure 28 we plot the robot's trajectories when tracking the line $h_{ref} = 0$ (in green) with $\theta_0 = 45^\circ$, $L_2 = 10$, and for $L_1/L_3 = 4$ (cyan), $L_1/L_3 = 2$ (red), and $L_1/L_3 = 4/3$ (blue). For all trajectories, the observed values for the design parameters are $L_2^{obs} \cong 9$, and $(L_1/L_3)^{obs} \cong 3.2$ (cyan), $(L_1/L_3)^{obs} \cong 1.8$, $(L_1/L_3)^{obs} \cong 1.2$, respectively. Also, taking into account the input angle $\theta_0 = 45^\circ$, the damped sinusoidal model predict $L_1 = 1.75$ (cyan), $L_1 = 2.31$ (red), and $L_1 = 2.77$ (blue), while the observed values are $L_1^{obs} \cong 1.5$, $L_1^{obs} \cong 1.9$ and $L_1^{obs} \cong 2.3$ respectively. As expected, these values are slightly smaller than the ones predicted by the damped sinusoidal trajectory.

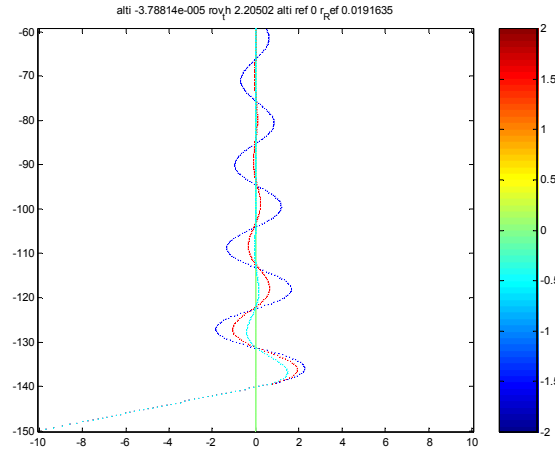


Figure 28: Robot's trajectories when tracking the line $h_{ref} = 0$ (in green) with $\theta_0 = 45^\circ$, $L_2 = 10$, and for $L_1/L_3 = 4$ (cyan), $L_1/L_3 = 2$ (red), and $L_1/L_3 = 4/3$ (blue).

The reference signal r_{ref_k} generated in the previous example for tracking the $h_{ref} = 0$ contour line is shown in Figure 29, where the colors are the same as in the corresponding trajectories. Note the smoothness of the generated commands.

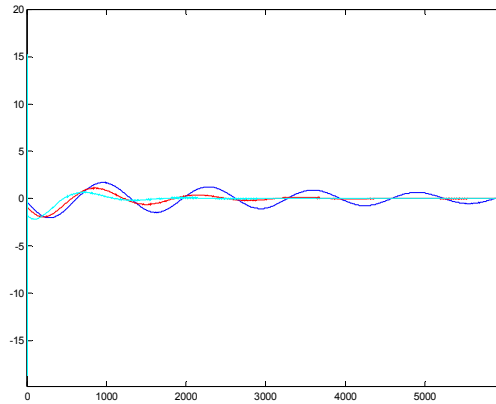


Figure 29: Generated command r_{ref} when tracking the line $h_{ref} = 0$ (in green) with $\theta_0 = 45^\circ$, $L_2 = 10$, and for $L_1 / L_3 = 4$ (cyan), $L_1 / L_3 = 2$ (red), and $L_1 / L_3 = 4/3$ (blue).

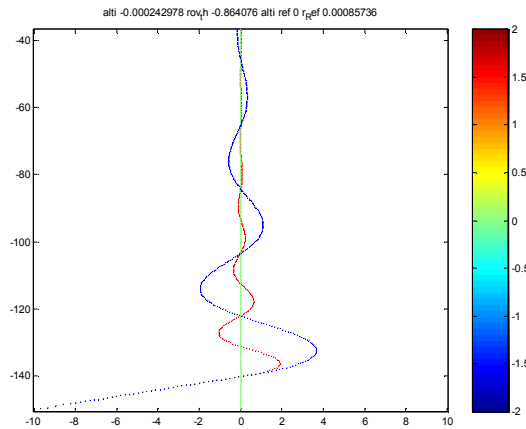


Figure 30: Robot's trajectories when tracking the line $h_{ref} = 0$ (in green) with $\theta_0 = 45^\circ$, $L_1 / L_3 = 2$, for $L_2 = 10$ (red), and for $L_2 = 20$ (blue)

In Figure 30, we observe the influence of L_2 in the robot's trajectory. Remark that, as predicted by the damped sinusoidal model, L_1 is proportional to L_2 for constant parameters L_3 / L_1 and θ_0 .

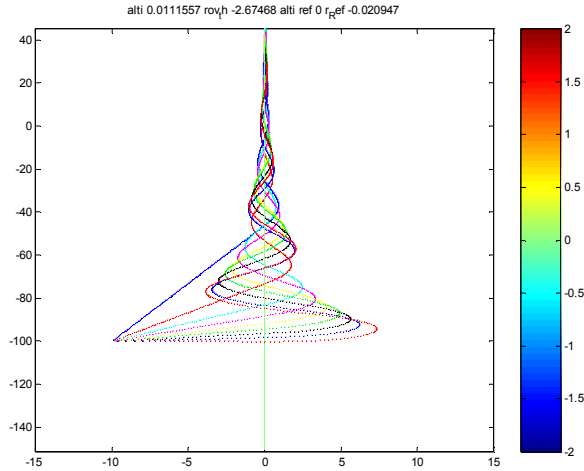


Figure 31: Robot's trajectories as a function of θ_0

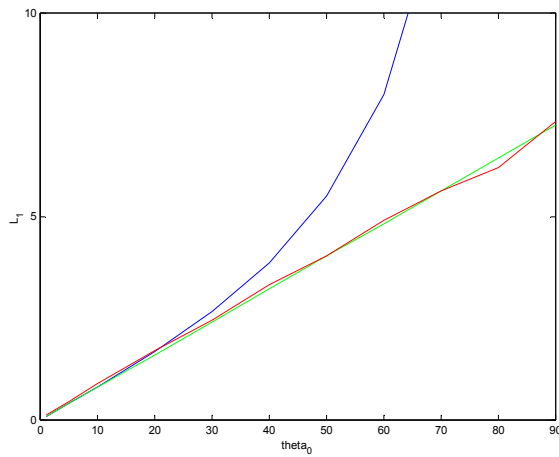


Figure 32: Evolution of L_1 as a function of θ_0

Considering $L_1/L_3 = 2$, and $L_2 = 20$, we plot in Figure 31 the trajectories described by the robot for $\theta_0 = 10^\circ, \dots, 90^\circ$ with a step of 10° , and in Figure 32 the evolution of L_1 , as a function of θ_0 , predicted by the damped sinusoidal model (blue), obtained by simulation (red), and the result of assuming a linear variation with θ_0 (green) instead of a linear variation with $\tan \theta_0$, i.e.

$$L_1 \cong \frac{\theta_0}{\sqrt{\alpha^2 + f^2}} e^{-\frac{\alpha}{f} \arctan\left(\frac{f}{\alpha}\right)}.$$

This result, which still needs to be better studied, allows us to better characterize the behaviour of the contour tracking system.

Simulation results

Now let us consider the more realistic scenario represented in Figure 33, where the sea bottom relief is modeled by the superposition of Gaussian functions. Moreover, the slope of the terrain is estimated on-line, and its value used to automatically fix the controller gains K_p and K_D .

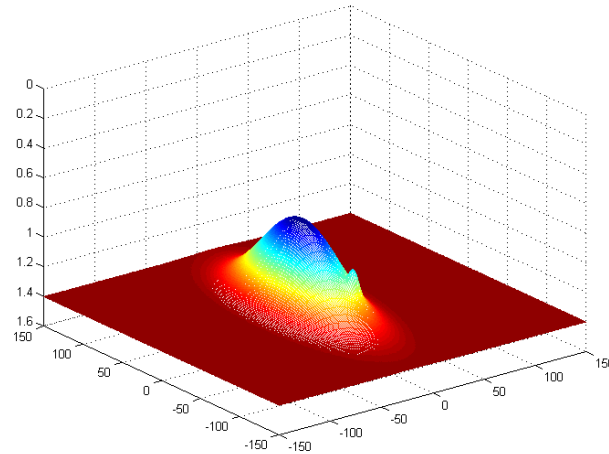


Figure 33: simulation scenario

Considering $L_1 / L_3 = 2$, we plot in Figure 34 and 35, respectively, the trajectories and the reference signal r_{ref} obtained for $L_2 = 5$ (black), $L_2 = 10$ (blue), $L_2 = 20$ (red), and $L_2 = 30$ (magenta). Remark that, in one hand L_2 characterizes the frequency of oscillation of the trajectory around the contour line, but in the other hand it is also directly proportional to the amplitude of the oscillations. Consequently, iso-depth lines with high curvature require a small L_2 to be followed with a small error. However, the improvement in the system agility obtained by reducing L_2 leads to larger variations of the command signal.

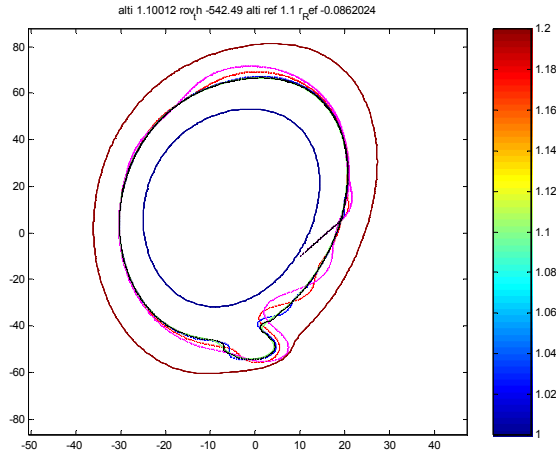


Figure 34: Robot's trajectories as a function of L_2

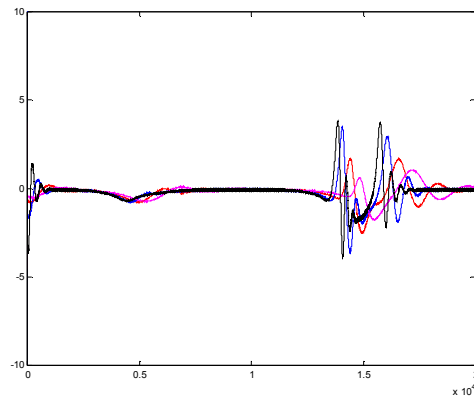


Figure 35: Command signal r_{ref} as a function of L_2

In Figure 36 and 37, we show the behaviour of the tracking system with respect to changes in L_1/L_3 . We fix $L_2 = 10$, and $L_1/L_3 = 2$ (blue), $L_1/L_3 = 8/5$ (red), and $L_1/L_3 = 4/3$ (magenta). Remark that, L_1/L_3 establishes the damping observed in the transitory behaviour of the robot's trajectory. Along with L_2 it indicates the spatial duration of the transient behaviour of the trajectories when the robot deflects from the iso-depth line being tracked.

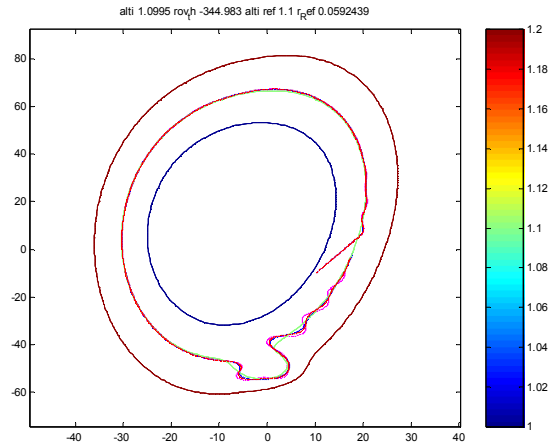


Figure 36: Robot's trajectories as a function of L_1 / L_3

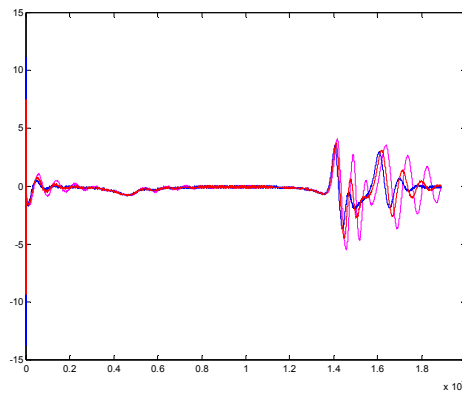


Figure 37: Command signal r_{ref} as a function of L_1 / L_3 .

Future work

The previous sections of this Chapter describe the work done so far concerning the definition of contour-tracking algorithms using altitude measures. The actual

implementation of these algorithms is dependent on successful solution of a number of issues:

- i. as the preceding section shows, the natural sensor for defining a contour tracking task is a rate gyro (measuring the instantaneous heading rate of the platform). This sensor is not currently present in the MAUVE platform. Its physical and electronic installation has been foreseen to occur in the beginning of 2003, followed by immediate software integration (modification of the UCV code for sensor acquisition and integration of the low level rate controller and contour tracker).
- ii. our simulation studies assume that the rate controller is implemented at 10 Hz. This requires that the altitude measures be available at the same rate. In the present MAUVE architecture, the altimeter measures are acquired by the MMNS, which runs at a 1 Hz cycle. Achievement of the target control rate of 10 Hz requires that an additional physical connection be established between the UCV board (where the rate controller runs) and the altimeter.
- iii. The simulations presented consider moderate error levels due to pitch oscillations and altimeter noise. The actual evaluation of these noise levels may lead to the need to consider prior filtering of the altimeter measures, which has not been considered yet. We are waiting that extensive trials of the complete MAUVE system be made (providing altimeter measures in the Bay of Villefranche-sur-mer, whose bathymetry is known) to assess this issue.

Conclusions

In this deliverable we present work done in the Sumare project concerning the definition of contour tracking behaviours for the two underwater platforms operated in the project: the ROV Phantom and the AUV Mauve, addressing milestones Mil 4.1 and 4.2 of the project. Each platform uses distinct perceptual modalities: video data for the Phantom and acoustic altitude measures for MAUVE. Results of real at-sea experiments are presented, illustrating the performance of the visual tracker for a specific case of a transition between Posidonia and sand, while the performance of the altitude tracker is shown only in simulated data.

Two distinct situations have been identified in the project concerning *visual tracking of boundaries*: existence of a neat transition between distinct benthic regions, and the case when a gradual transition region can be observed between more or less homogenous regions. Actual implementation of a visual tracker has been achieved for the first case (with complete integration in the Phantom software architecture). The second situation has been only formally studied, in the context of a PhD thesis conducted in the framework of the project [rolfes2003], and only results under simulated scenarios are available at the moment.

Two different versions of *contour tracking using acoustic measures* have been identified in the project workprogram: use of altimeter data (fixed single acoustic beam, solidary with the robot's body, and providing direct measures of the altitude above sea floor) and of sonar data (scanning acoustic beam providing, for each look direction, the complete response of the surrounding medium – a profile). Only the first configuration is addressed in this deliverable, which proposes a control architecture and associated algorithms for its implementation in the MAUVE platform. The other situation, which will enable us to track lines of transition between distinct sea bottom habitats on the basis of acoustic data (as a possible complement or alternative to the use of visual information) is not addressed here. Promising preliminary studies on the segmentation of sonar profiles corresponding to distinct sea bed types (more precisely, sand and Posidonia) have already been obtained, which will be reported in another deliverable of the project. The definition of a tracker based on this kind of *on/off* information has not been attempted yet. It will be the subject of subsequent studies, whose results will be reported in deliverable D4.2.

Bibliography

- Rolfes2002 Stefan Rolfes, *Stochastic Geometry: an approach to featureless perception-based robot navigation*, Thèse de Doctorat, Université de Nice Sophia Antipolis, December 2002.
- RolfesRendas2002 Stefan Rolfes and Maria-João Rendas, *Statistical environment representation for navigation in natural environments*, Robotics and Autonomous Systems 41 (2002) 129-136, Elsevier.
- BaratRendas2002 Christian Barat and Maria-João Rendas, *Classification of sonar measures using optimized wavelets*, Proc. of OCEANS'02, Biloxi, Mississippi, 2002.
- RendasRolfes2002 Maria-João Rendas and Stefan Rolfes, *Underwater robot navigation using benthic contours*, Proc. of OCEANS'02, Biloxi, Mississippi, 2002.
- RolfesRendas2002a Stefan Rolfes and Maria-João Rendas, *Mobile Robot Localization based on Random Closed Set Model Maps*, Proc. ICRA 2002, Washington D.C., USA, 2002.
- TenRenFol2001 Albert Tenas, Maria-João Rendas and Jean-Pierre Folcher, *Image segmentation by unsupervised adaptive clustering in the distribution space for AUV guidance along sea-bed boundaries using vision*, Proc. of OCEANS'01, Hawai, USA, 2001.