Sensor-Based Control of Holonomic Autonomous Underwater Vehicles

Aristide Santos, Bernard Espiau, Patrick Rives, Daniel Simon, Vincent Rigaud

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Sensor-Based Control of Holonomic Autonomous Underwater Vehicles

Aristide Santos, Bernard Espiau, Patrick Rives, Daniel Simon, Vincent Rigaud

N° 2609

Juillet 1995

PROGRAMME 4

ISSN 0249-6399

INRIA
SOPHIA ANTIPOLIS

Sensor-Based Control of Holonomic Autonomous Underwater Vehicles

Aristide Santos*, Bernard Espiau**, Patrick Rives***, Daniel Simon****, Vincent Rigaud*****

Programme 4 — Robotique, image et vision
Projet ICARE

Rapport de recherche n°2609 — Juillet 1995 — 39 pages

Abstract: For underwater vehicles to be self-sufficient in an *a priori* unknown environment, feedback from the environment through altitude sensors is essential. This paper presents how this feedback can be efficiently achieved for fully actuated holonomic autonomous underwater vehicles (AUVs) in which different simultaneous primary tasks have to concur to the achievement of a mission. These vehicles are submitted to constrained motion due to the operational requirements of the load sensor(s) that is(are) carried for a mission, and are equipped with acoustic range altitude sensors. The approach relies on the task-function concept which is based on the reduction of the control problem to the regulation to null value of a task-function defined by the user. Two ways of taking into account different primary tasks in the determination of the task-function to be regulated to zero are presented : a minimization of a cost function and a smooth switching between tasks. The presented control law appears to be robust against sensory data noise and efficient for the following of an a priori unknown and profile-varying environment.

Key-words: sensor-based control, autonomous underwater vehicles (AUV), *a priori* unknown environment, acoustic sensors, task-function approach, obstacle avoidance, contour following.

(Résumé : tsvp)

This work is supported by Ifremer and by the PACA Region (FRANCE), and is part of the EEC MAST II program under grant CT92-0028

*E-mail : santos@sophia.inria.fr
**INRIA – Rhône Alpes, 46 avenue Félix Viallet, 38031 Grenoble Cedex, FRANCE. E-mail : Bernard.Espiau@imag.fr
***E-mail : rives@sophia.inria.fr
****E-mail : dsimon@sophia.inria.fr
*****Ifremer – Centre de Toulon, Zone Portuaire de Brégaillon, BP 330, 83507 La-Seyne-sur-mer Cedex, FRANCE. E-mail : rigaud@toulon.ifremer.fr

Commande Référencée Capteurs des Véhicules Sous-Marins Autonomes Holonomes

Résumé : Rendre des véhicules sous-marins autonomes dans un environnement *a priori* inconnu requiert l'utilisation de données sensorielles issues de capteurs permettant d'avoir une perception, si locale soit-elle, de cet environnement. Nous présentons ici comment ces données peuvent être introduites de manière satisfaisante dans une boucle de commande dans le cas des véhicules autonomes sous-marins holonomes (AUVs) pour lesquels différentes tâches primaires doivent être prises en compte simultanément pour permettre la réalisation d'une mission donnée. Ces véhicules sont soumis à des contraintes de déplacement dues aux contraintes opérationnelles des charges utiles transportées pour la mission. L'approche proposée repose sur le concept de la fonction de tâche, qui réduit le problème de commande à la régulation à zéro d'une fonction de tâche définie par l'opérateur. Deux manières de prendre en compte différentes tâches dans une même boucle de commande sont présentées : la première repose sur la minimisation d'une fonction coût, et la seconde sur un "switching" continu entre les tâches primaires à considérer. La commande qui en résulte s'avère être robuste aux données sensorielles bruitées, et respecte les objectifs fixés pour le suivi de fond de mer de profil variable et *a priori* inconnu.

Mots-clé : commande référencée capteurs, véhicules autonomes sous-marins (AUVs), environnement *a priori* inconnu, capteurs acoustiques, approche fonction de tâche, évitement d'obstacle, suivi de contour continu.

1 Introduction

Following the human-piloted and remotely operated ones, the third generation of underwater vehicles is expected to be entirely self-sufficient. Application fields of such vehicles (AUVs) are as appealing as numerous : scientific (geology, geophysics, ...), environmental (waste disposal monitoring, ...), or commercial (oil and gas, submarine cables, ...) [Masson et al., 1994]. During an operation, an AUV is supposed to carry load sensors (sidescan sonar, bathymetry, sub-bottom profiler...) and to track a certain type of trajectory (e.g. either travelling from point A to point B or performing the systematic exploration of a zone along a grid). These load sensors induce strong constraints on the vehicle motion [Ageev, 1994, Collar et al., 1994]. Furthermore, many of them require to follow an *a priori* unknown and profile-varying sea bottom at a desired reference vertical altitude, while maintaining a nominal constant axial speed, and avoiding obstacles when necessary. For that purpose, AUVs are equipped with altitude sensors which provide a local perception of the environment. This perception consists of distance measurements from sensors with respect to the environment. The sensors are assumed to be rigidly linked on the vehicle body. The raised question then is : how to use them inside a real-time continuous closed-loop control law in order to correctly and safely achieve an assigned mission ?

The integration of sensory data in a closed-loop control of mobile robots or vehicles has known several developments these last years. Two large classes of approaches can be distinguished :

- the *behavioral approach*, consisting in directly relating inputs (or *stimuli*) to outputs (or *actions*) through systems state [Brooks, 1989, Soldo, 1990]. The specificity of the input-output linkage is however too restrictive for a general use.

- the *sensor-based control approach*, which directly introduces the sensory data in the robot closed-loop control. The main benefit consists in specifying the servoing problem directly in terms of a regulation on sensory data. Therefore, the 2D/3D reconstruction of the scene feature is generally not required. Several concepts exist including [Zapata and Lépinay, 1991] which presents an algorithm based on the detection of the deformations of a Deformable Virtual Zone (DVZ) surrounding the vehicle, or [Holenstein and Badreddin, 1991] which presents an approach relying on the concept of Artificial Potential Field built around the vehicle. Nevertheless, the *task-function* concept [Samson et al., 1990] offers an even more global approach for both the minimal altitude sensors configuration and the high level control approach. Indeed, the control problem is reduced to the simple regulation to null value of a user-defined task-function (see Figure 1). Moreover, taking into account the analogy between vir-

tual solid contacts and the linkages one can build between the sensors system and the sensible environment, a minimal configuration for the sensors system can be deduced for any task affected to the vehicle.

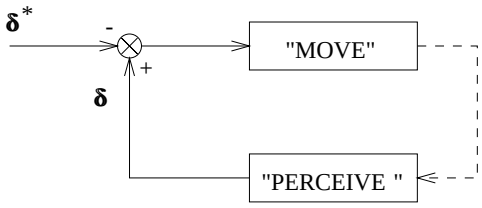


Figure 1: Sensory data servoing

We develop through this paper a sensor-based control law for a class of fully-actuated holonomic[1] vehicles that are assumed to be controllable through their velocity screw, meaning that dynamical effects are not taken into account. These vehicles are intended to follow an *a priori* unknown and profile-varying sea bottom at a nominal advancement velocity u_0. The bottom following problem may be formulated as following :

1. considering the ideal path obtained at the desired constant reference altitude from the bottom, to make the vehicle align itself and follow it, taking into account the possible abrupt changes of the bottom profile ;

2. to have the vehicle's pitch angle correspond to the inclination of its orthogonal projection on the path.

Our objective in this paper is to show the convenience of the *task-function* approach when dealing with different tasks in an *a priori* unknown and changing environment, such as the deep underwater one.

The structure of the paper is as follows. In Section 2 we present the mathematical concept of the *task-function* approach, which is the the basis for design. Section 3 is devoted to its application to AUVs equipped with altitude sensors. Section 4 describes the concept of the minimization of a cost function when dealing with several different tasks, while Section 5 details the concept of a smooth switch between tasks. A sensor-based control law is presented in Section 6 the performances of which are illustrated in Section 7. Section 8 then concludes.

[1] According to [Shabana, 1989], a mechanical system is said to be holonomic when all the kinematic constraints are fully integrable. All other systems are obviously nonholonomic.

2 The task-function concept

2.1 Some mathematical notations

Consider $\mathcal{E}$, the three-dimensional affine Euclidean space, the related vector space being $\mathbb{R}^3$. The configuration space of a rigid body, which is also the frame configuration space, is the Lie group of displacements of $\mathcal{E}$, or $\mathcal{SE}(3)$, isomorphic to $\mathbb{R}^3 \times \mathcal{SO}(3)$, where $\mathcal{SO}(3)$ is the group of rotations. It is a six-dimensional differential manifold an element of which is called a situation (i.e. position and attitude) and is denoted as $\mathbf{r}$. The tangent space to $\mathcal{SE}(3)$ at identity, denoted se_3, is a Lie algebra isomorphic to the Lie algebra of equiprojective fields of $\mathcal{E}$ in $\mathbb{R}^3$. That means that any element of se_3 can be represented by a velocity screw. A screw $\mathbf{H}(\mathbf{H}(O), \mathbf{u})$ is defined by its vector $\mathbf{u}$ and the value of its field $\mathbf{H}(O)$ at a point O of $\mathcal{E}$.

Recalling that the screw product, which is a bilinear mapping, induces an isomorphism between a screw space and its dual (or cotangent space), the dual of se_3, denoted se_3^*, is itself a screw space. The interaction screw, basic notion of the task-function approach, belongs to this space se_3^*.

2.2 General formalism

Let's consider $\boldsymbol{\delta} = (\delta_1, \ldots, \delta_j, \ldots, \delta_p)^T$, set of p elementary sensory signals δ_j, $j = 1..p$, which are assumed :

1. to depend only on the relative situation (i.e. location and attitude) between the sensors and the environment,
2. and to be a $\mathcal{C}^2$ function with values in $\mathbb{R}$ and domain in $\mathcal{SE}(3)$.

Taking into account these assumptions, δ_j for each sensor j then admits a derivative with respect to time t represented by an element of se_3, or a screw, we could express through :

$$\dot{\delta}_j = \frac{\partial \delta_j}{\partial \bar{\mathbf{r}}} \frac{\partial \bar{\mathbf{r}}}{\partial t} = \mathbf{H}_j \bullet \mathbf{T}_{ST} \tag{1}$$

where

- "$\bullet$" is the screw product operation. For any considered point O and two screws, $\mathbf{H}_1(\mathbf{H}_1(O), \mathbf{u}_1)$ and $\mathbf{H}_2(\mathbf{H}_2(O), \mathbf{u}_2)$, it is defined by :

  $$\mathbf{H}_1 \bullet \mathbf{H}_2 = < \mathbf{u}_1, \mathbf{H}_2(O) > + < \mathbf{u}_2, \mathbf{H}_1(O) >$$

  where $< ., . >$ is the usual scalar product.

- $\bar{\mathbf{r}}(t)$ is the *relative* AUV–environment situation.

- $\mathbf{H}_j$ is the interaction screw that entirely characterizes both the variation of the data δ_j and the interaction between the sensor and its environment.
- $\mathbf{T}_{ST}$ is the velocity screw of the relative displacement between the sensor and the target.

For a given relative situation $\bar{\mathbf{r}}$, it is important to notice that $\mathbf{H}_j$ only depends on :

1. the relative position between the sensor and the target,
2. the sensor physical characteristics,
3. and is independent of the vehicle body that is supposed to be rigidly linked to the sensor.

Referring to equation (1), the set of motions or velocity screws $\{\mathbf{T}^*\} = \{(\mathbf{V}, \boldsymbol{\omega})^*\}$ that leaves the whole set $\boldsymbol{\delta}$ invariant from an initial relative position of reference, is characterized by the reciprocal space $\mathcal{T}$ to the subspace generated by the set $\{\mathbf{H}_1, \ldots \mathbf{H}_j, \ldots \mathbf{H}_p\}$ in the six-dimensional vector space of screws.

From [Samson et al., 1990], it follows that the formalism of screws can also be used to describe contacts between rigid bodies, leading to a possible parallelism between true contacts and task-functions. The underlying idea is the following :

> Imposing the condition $\dot{\delta}_j = 0$ on every sensor (S_j), $j = 1..p$, is equivalent to introducing constraints in the configuration space $\mathcal{SE}(3)$ in order to achieve a *virtual contact* between the objects and the body carrying the sensors.

From a geometrical point of view, $\mathcal{T}$ is entirely defined by the kernel of the following matrix $\mathbf{L}^T$ with dimension $p \times n$:

$$\mathbf{L}^T = \begin{bmatrix} \mathbf{u}_1^T & \mathbf{H}_1^T(P) \\ \vdots & \\ \mathbf{u}_j^T & \mathbf{H}_j^T(P) \\ \vdots & \\ \mathbf{u}_p^T & \mathbf{H}_p^T(P) \end{bmatrix} \tag{2}$$

evaluated at some point P, with vectors expressed in the same basis. The matrix $\mathbf{L}$ is called the *interaction screw*, and the relation (1) becomes :'

$$\dot{\boldsymbol{\delta}} = \mathbf{L}^T \, \mathbf{T}_{ST} \tag{3}$$

It can be shown that the class of the virtual linkage is the number of degrees of freedom of the body that leave the outputs of all sensors invariant. The class of the virtual linkage is given by :

$$N \;=\; n - rank(\mathbf{L}), \quad \text{with } rank(\mathbf{L}) \leq \min(n,p) \tag{4}$$

where n is the total number of degrees of freedom of the body that can be constrained by any task.

We then notice that there exists a minimal number p_m achieving a given N such that $p_m = rank(\mathbf{L})$. p_m corresponds to the *minimal number of sensors* required for the achievement of a given type of task affected to the vehicle. In the case where $p > p_m$, the notion of *redundancy* is introduced. In practice, this is often sought-after for a robust behavior and the safety of the vehicle when in mission. Study of optimal, redundant or not, altitude sensors architecture [Santos and Simon, 1994] has led to geometrical results depending on altitude sensors physical modelling (ray, wide-beam sensor, or Kuc's model [Kuc, 1990]. Nevertheless, the real physical model of an acoustic altitude sensor is not easy to operate with [Kuc and Siegel, 1987, Budenske and Gini, 1994]. In order not to be device-dependent, we will introduce the notion of *Relevant Information* (RI). Indeed, the results we present remain valid when considering a virtual sensor that provides us with a distance, called the Relevant Information. This virtual sensor may be in practice a combination of several sensors, sometimes of different types, the information of which are filtered in such a manner that the RI is obtained. Several studies exist on such methods for efficient sensory data merging [Rigaud, 1990, Korba, 1994, Kuc and Bozma, 1991].

In the general task-function framework [Samson et al., 1990], the control problem is considered to be well-posed when it exists a n-dimensional function, with some regularity properties, that vanishes when the user's objective is completed. Here, n is the number of degrees of freedom of the vehicle (without any constrained axis, $n = 6$ for AUVs). For finding such a function, we firstly define a task, $\mathbf{e}_1$, considered as *having priority* with dimension $m \leq n$. For control purposes, $\mathbf{e}_1$ should be $\mathcal{C}^2$ except on a finite number of points. Then, a *secondary* admissible task, $\mathbf{e}_2$, can be set as representing the minimization of a scalar cost function h_s, with gradient $\mathbf{g}_s = \frac{\partial h_s}{\partial \mathbf{r}}$, under the constraint of perfect regulation of $\mathbf{e}_1$. A general task-function, or *hybrid task*, achieving the goal of the minimization of h_s under the constraint $\mathbf{e}_1 \;=\; \mathbf{0}$ can finally be defined as :

$$\mathbf{e} \;=\; \mathbf{W}^+ \, \mathbf{e}_1 \;+\; \alpha \, \left(\mathbf{I}_n \;-\; \mathbf{W}^+ \, \mathbf{W} \right) \, \mathbf{g}_s \tag{5}$$

where

- $\mathbf{W}$ is a full-rank matrix such that :

$$Ker\,(\mathbf{W}) \;=\; Ker\,(\mathbf{J}_1) \tag{6}$$

or equivalently

$$R(\mathbf{W}^T) = R(\mathbf{J}_1^T), \quad \text{with } \mathbf{J}_1 = \frac{\partial \mathbf{e}_1}{\partial \mathbf{r}} \tag{7}$$

where $R(.)$ stands for "*Range*". It fully characterizes the constrained axes of the vehicle.

- $\mathbf{W}^+$ is the pseudo-inverse of $\mathbf{W}$.
- α is a positive weighting scalar.
- $(\mathbf{I}_n - \mathbf{W}^+\mathbf{W})$ is the orthogonal projection operator onto the null space of $\mathbf{W}$, with $\mathbf{I}_n$ representing the $n \times n$ identity matrix.

In the case where sensory data are involved, $\mathbf{e}_1$ can be defined as :

$$\mathbf{e}_1 = \mathbf{C}\ \mathbf{e}_p(\mathbf{r}, t) \tag{8}$$

where

- $\mathbf{C}$ with dimension $m \times p$, called the *combination matrix*, allows us to consider a number of sensory data greater than the number of degrees of freedom of the vehicle. A possible choice for $\mathbf{C}$, which has interesting consequences for the robustness of the control [Samson et al., 1990], is :

$$\mathbf{C} = \mathbf{W}\ \mathbf{L}, \quad \text{with } \mathbf{L}^T = \frac{\partial \mathbf{e}_p}{\partial \mathbf{r}} \tag{9}$$

  Note that, for simplifying the computation of the derivatives, we will assume in all the following that $\mathbf{L}$ is locally constant or only time-varying.

- $\mathbf{e}_p(\mathbf{r}, t)$ is a p-dimensional vector based on sensory data that is to be regulated to $\mathbf{0}$.

The trajectory $\mathbf{r}^*(t)$ solution of $\mathbf{e}(\mathbf{r}, t) = \mathbf{0}$ is called the *ideal trajectory*.

3 Application to underwater vehicles

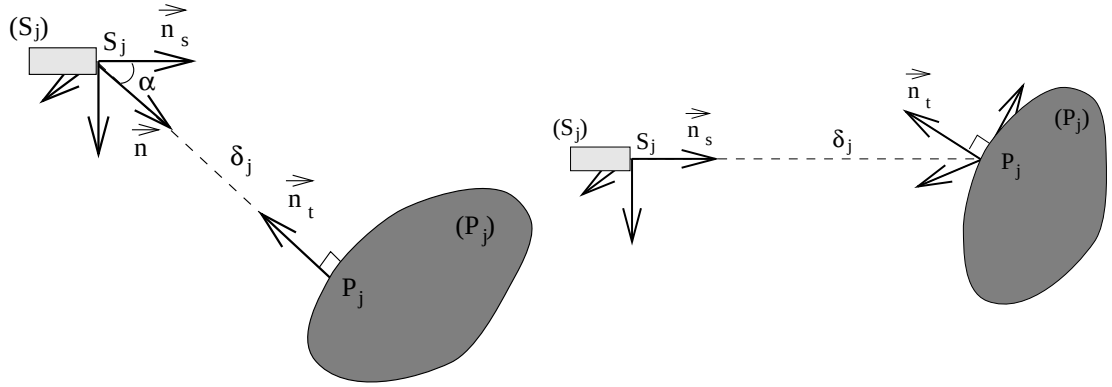
Building a task-function for autonomous vehicles using altitude sensors can be done considering three steps :

1. the determination of the nature of the interaction between the sensor and the environment,
2. the determination of the corresponding altitude sensors system configuration,
3. the determination of the task(s) to be achieved.

After the task-function vector has been built for a given mission, it remains to determine the control law to regulate it to zero.

3.1 Modelling of the sensor-environment interaction

A convenient model simple enough for calculations is the geometric model of a non-null beam width angle altitude sensor as depicted on Figure 2-(a).



(a) sensor with non null beam width angle (b) sensor with null beam width angle

Figure 2: Geometrical modellings of an altitude sensor

Let us consider $s_j = f(\delta_j)$, the sensory information given by the sensor (S_j) with the frame F_{S_j} attached to it and origin S_j. It can be shown [Samson et al., 1990, Santos and Simon, 1994] that the variation of s_j can be expressed through :

$$\begin{aligned} \dot{s_j} \;=\; & < \left(\frac{\partial f}{\partial \delta_j} \;+\; \frac{1}{\delta_j \;\tan(\alpha)} \; \frac{\partial f}{\partial \alpha} \right) \; \mathbf{n}_t \;+\; \frac{1}{\delta_j \;\sin(\alpha)} \; \frac{\partial f}{\partial \alpha} \; \mathbf{n}_s, \; \mathbf{V}_{S_j/F_{P_j}} > \\ & + \; < \; \frac{1}{\sin(\alpha)} \; \frac{\partial f}{\partial \alpha} \; (\mathbf{n}_s \;\wedge\; \mathbf{n}_t) \; , \; \boldsymbol{\omega}_{F_{S_j}/F_{P_j}} > \end{aligned} \tag{10}$$

where $< .,. >$ is the usual scalar product. The interaction screw is then obtained :

$$\left\{ \begin{array}{lcl} \mathbf{u}_j & = & \left(\frac{\partial f}{\partial \delta_j} \;+\; \frac{1}{\delta_j \;\tan(\alpha)} \; \frac{\partial f}{\partial \alpha} \right) \; \mathbf{n}_t \;+\; \frac{1}{\delta_j \;\sin(\alpha)} \; \frac{\partial f}{\partial \alpha} \; \mathbf{n}_s \\ \\ \mathbf{H}_j(S_j) & = & \frac{1}{\sin(\alpha)} \; \frac{\partial f}{\partial \alpha} \; (\mathbf{n}_s \;\wedge\; \mathbf{n}_t) \end{array} \right. \tag{11}$$

However it appears clearly that $\mathbf{n}_t$, α may be difficult to be exactly known in practice. When simplifying again the model by using the model with a null beam width angle (see Figure 2-(b)), the equation (10) becomes :

$$\dot{s_j} \;=\; -\; \frac{1}{< \mathbf{n}_t, \mathbf{n}_s >} \; \frac{\partial f}{\partial \delta_j} \; \left[< \mathbf{n}_t, \mathbf{V}_{S_j/F_{P_j}} > \;+ \delta_j \;\; < (\mathbf{n}_s \wedge \mathbf{n}_t), \boldsymbol{\omega}_{F_{S_j}/F_{P_j}} > \right] \tag{12}$$

The components of the interaction screw attached to this model of sensor is then :

$$\left\{ \begin{array}{rcl} \mathbf{u}_j & = & - \dfrac{1}{< \mathbf{n}_t, \mathbf{n}_s >} \dfrac{\partial f}{\partial \delta_j} \mathbf{n}_t \\ \mathbf{H}_j(S_j) & = & - \dfrac{\delta_j}{< \mathbf{n}_t, \mathbf{n}_s >} \dfrac{\partial f}{\partial \delta_j} (\mathbf{n}_s \wedge \mathbf{n}_t) \end{array} \right. \tag{13}$$

As the component $< \mathbf{n}_t, \mathbf{n}_s >$ cannot be measured, and assuming that the obtained signal is the distance δ_j, we will use in practice the following approximation :

$$\left\{ \begin{array}{rcl} f(\delta_j) & = & \delta_j \\ \alpha & \approx & 0 \\ \mathbf{n} & = & \mathbf{n}_s = -\mathbf{n}_t \end{array} \right. \tag{14}$$

The expression of the approximated model of interaction screw associated with the sensor (S_j) then becomes :

$$\text{For any point } Q \in (S_j),\ \mathbf{L}(Q) = \left\{ \begin{array}{rcl} \mathbf{u}_j & = & \mathbf{n}_t \\ \mathbf{H}_j(Q) & = & \mathbf{QP}_j \wedge \mathbf{u} \end{array} \right. \tag{15}$$

Remark For any point Q belonging to the sensor axis (S_j, P_j) and when $\mathbf{n}_s // \mathbf{n}_t$ as in model (14)–(15) above, we find that $\mathbf{H}_j(Q) = \mathbf{0}$, and therefore that the interaction screw is a slider through Q.

3.2 Admissible tasks, linkages — minimal number of sensors

3.2.1 Admissible tasks

Considering the missions that could be affected to it [Masson et al., 1994], an AUV should handle two different tasks that require the integration of sensory data in the closed-loop control :

T1 The *obstacle avoidance* when facing abrupt breaks of the bottom profile (e.g. vertical cliffs) ;

T2 The *contour following* when the bottom profile is smooth, with no break. We will distinguish two types of contour following : either linear (**T2a**) — a 2D problem, example of a long single profile survey — or of surface (**T2b**) — a 3D problem, example of a detailed grid survey of a zone —.

Before going any further, let precise the sense of "break". Consider that we want to maintain a moving vehicle at a constant altitude y_d with respect to the bottom, represented by the curve $(\mathcal{C}_b)$ that is characterized mainly by its curvature $c(s)$ (s denoting the curve curvilinear coordinate) [Samson, 1992].

Define $(\mathcal{C}_t)$, the trajectory induced by the following of the bottom profile at a desired reference altitude of y_d. It is assumed that the radius of any circle tangenting $(\mathcal{C}_t)$ at two or more points, and the interior of which does not contain any point of the curve, is bounded from below by some positive number denoted as r_{min} (see Figure 3). For example, if $(\mathcal{C}_t)$ is a straight line, then $r_{min} = \infty$; if $(\mathcal{C}_t)$ is a circle, then r_{min} is the circle radius; and for a corner, $r_{min} = 0$. We consider the bottom profile to be smooth, or with no break, when $y_d < r_{min}$. Similarly, the resulting path of the vehicle is considered as smooth when the current altitude y with respect to bottom is such that $y - y_d < r_{min}$.

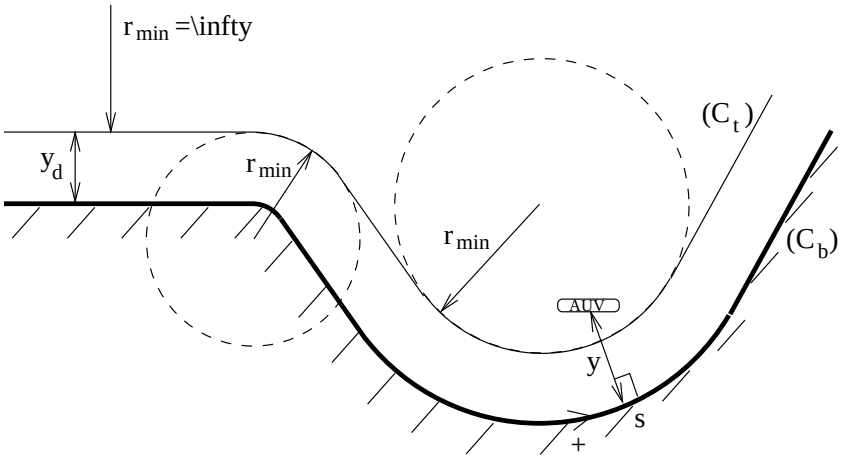


Figure 3: AUV–environment configuration

3.2.2 Linkages and canonical configurations

Making the analogy with the kinematics of solid contacts leads to three types of adequate linkages :

L1 a *point/plane* linkage, the class of which is C5 (see Figure 4-(a)). It requires *at least one (1) sensor* to be achieved. This linkage is attached to the obstacle avoidance task T1. The interaction screw corresponding to Figure 4-(a) calculated at point $O \equiv P_1$ is :

$$\mathbf{L}(O) = \begin{pmatrix} \mathbf{u} \\ \mathbf{H}(O) \end{pmatrix} = \begin{pmatrix} 0 & 0 & -1 & 0 & 0 & 0 \end{pmatrix}^T$$

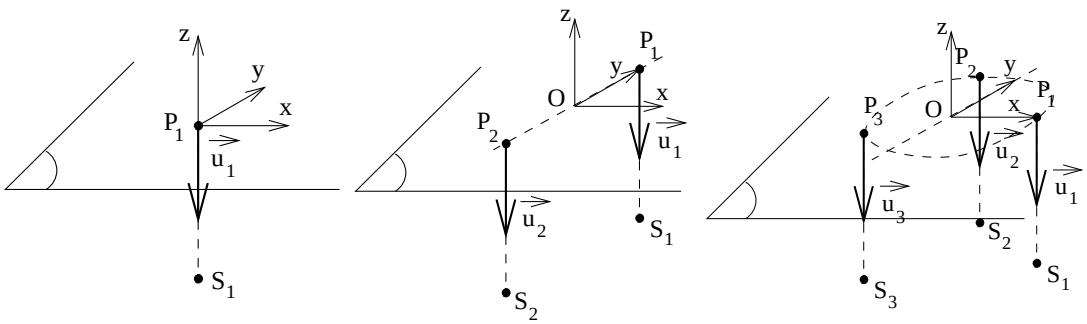
L2 a *line/plane* linkage, the class of which is C4 (see Figure 4-(b)). It requires *at least two (2) sensors* to be achieved. This linkage may be attached either to the linear contour following T2a, or to the obstacle avoidance task T1 in case of redundancy. The interaction screw corresponding to

Figure 4-(b) is :

$$\mathbf{L}(O) = \begin{pmatrix} \mathbf{u}_1 & \mathbf{u}_2 \\ \mathbf{H}_1(O) & \mathbf{H}_2(O) \end{pmatrix} = \begin{pmatrix} 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & -1 & 0 & 0 \end{pmatrix}^T$$

L3 a *plane/plane* linkage, the class of which is C3 (see Figure 4-(c)). It requires *at least three (3) sensors* to be achieved. This linkage may be attached to the contour following of surface T2b. In the canonical example of Figure 4-(c) where the points P_1, P_2, P_3, are equidistant on a circle of radius unity centered on O, the corresponding interaction screw is :

$$\mathbf{L}(O) = \begin{pmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \mathbf{u}_3 \\ \mathbf{H}_1(O) & \mathbf{H}_2(O) & \mathbf{H}_3(O) \end{pmatrix} = \begin{pmatrix} 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & -1 & -\sqrt{3}/2 & -1/2 & 0 \\ 0 & 0 & -1 & \sqrt{3}/2 & -1/2 & 0 \end{pmatrix}^T$$



(a) the point/plane linkage (b) the line/plane linkage (c) the plane/plane linkage

Figure 4: Canonical configurations of linkage

We have defined above the *minimal number of sensors* according to the basic tasks that can be affected to an AUV. From the corresponding basic linkages, quite complex configurations can be built, that correspond to any primary or complex task for a given mission. Some examples are given in [Santos and Simon, 1994], and more complete analyses are done including sea bottom following associated with obstacle avoidance and slope ascents/descents.

3.3 Primary task-functions definition

3.3.1 The obstacle avoidance task

We search for a function that is $\mathcal{C}^2$ except on a finite number of points, and that can generate repulsive action when facing bottom profile breaks. Any function $e_p^{[1]}$ achieving the conditions (i)–(iv) below is convenient :

(i) $e_p^{[1]} = f(\delta) \leq 0,\ \delta \in]0, \delta_m]$ (a repulsive action is required)

(ii) $\frac{\partial e_p^{[1]}}{\partial \delta} \geq 0,\ \delta \in]0, \delta_m]$

(iii) $e_p^{[1]} = 0,\ \delta > \delta_m$

(iv) $e_p^{[1]}$ of class $\mathcal{C}^2$ on $]0, \delta_{max}]$

with the obvious condition : $0 < \delta_m \leq \delta_{max}$. δ is the distance measurement obtained from one sensor (S), δ_{max} represents the maximum sensor range, and δ_m defines the distance under which the obstacle avoidance has to be taken into account.

An example of such a function is :

$$e_p^{[1]}(\delta, \delta_m) = \begin{cases} -k_1 \left(\frac{\delta_m}{\delta} - 1 \right), & \text{when } \delta \leq \delta_m \\ 0, & \text{when } \delta > \delta_m \end{cases} \tag{16}$$

where k_1 may be a strictly positive increasing function of the ratio $\frac{u_0}{u_m}$, u_0 and u_m representing the constant nominal and maximum advancement velocities of the vehicle respectively.

It can be easily shown that the definition of this task-function includes the notion of protection areas (here reaction and perception areas) around the vehicle, as depicted on Figure 5.

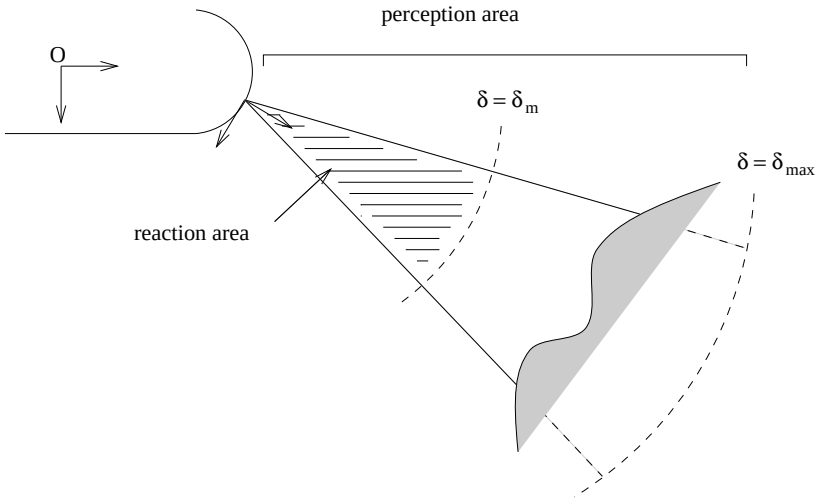


Figure 5: Protection areas defined in case of obstacle avoidance task

3.3.2 The contour following task

The task-function should be zero when the desired reference signal is obtained. A simple function is :

$$e_p^{[2]}(\delta, \delta^*) \;=\; k_2 \;(\delta \;-\; \delta^*)\,, \quad k_2 \;>\; 0, \quad \delta^* < \delta_{max} \tag{17}$$

where δ^* is the desired reference distance measurement from the sensor, δ_{max} is the maximum sensor range. k_2 may be defined as k_1 above.

Again in this case, it is shown that the protection areas (repulsion, attraction, perception areas) are implicitly taken into account through this task-function, as depicted on Figure 6.

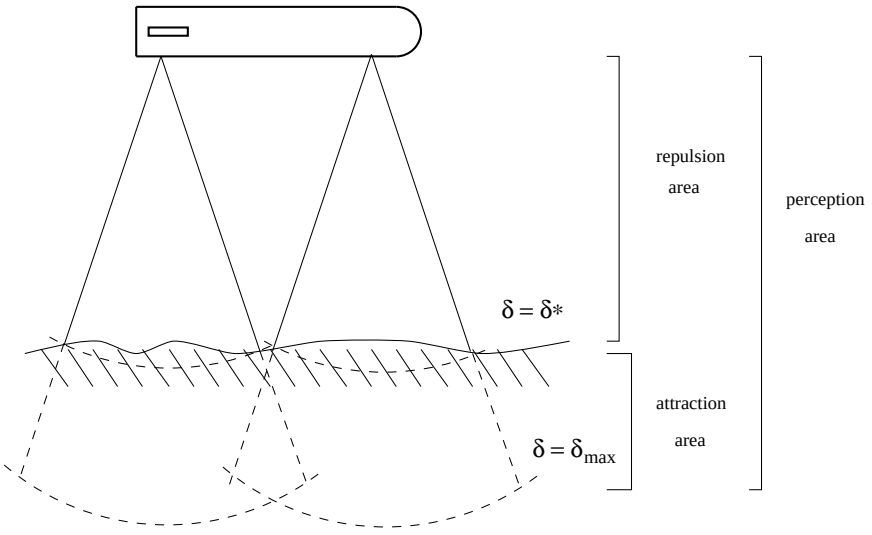


Figure 6: Protection areas defined in case of contour following task

3.3.3 Combining the Obstacle Avoidance and Contour Following tasks

In order to cope with a dynamically changing sea bottom, including smooth curves and abrupt corners and edges, both obstacle avoidance and contour following tasks should be permanently active during an AUV's mission.

A first simple approach is to merge the two primary task-functions (16) and (17) into a single one through a simple addition operation. Since the contour following task-function (17) may be positive or negative, this does not ensure that the resulting primary task-function is zero *if and only if* both tasks are achieved even if, thanks to a sufficient weighting, it is possible to ensure avoidance of fatal collision [Santos et al., 1994]. However, further simulations show that the behavior of the AUV during transient phases involving bottom profile breaks (corners or edges) highly depends on the weighting gains of the

task-functions. If these are high due to robustness requirements, an important variation on the global task-function error can lead to a brutal undesired response on the vehicle mechanics, and therefore damage actuators.

We present above in Section 4 and Section 5 two other more promising approaches. The first one, presented in the following Section 4, uses the principle of the minimization of a cost function based on the primary task-functions defined in (16)-(17), while the second one, detailed in Section 5, prefers to smoothly switch from one task to another. These two approaches are compared in Section 7 through simulation results using different bottom profiles including corners and edges.

4 Minimization of a cost function

We would like to ensure that both obstacle avoidance and contour following task-function vectors, $\mathbf{e}_1^{[1]}$ and $\mathbf{e}_1^{[2]}$ respectively, are $\mathbf{0}$ *if and only if* the final global task-function, $\mathbf{e}_1$ to be defined next, is $\mathbf{0}$. $\mathbf{e}_1^{[1]}$ and $\mathbf{e}_1^{[2]}$ are defined as in (8) using $\mathbf{C}_1$ and $\mathbf{C}_2$, as defined in (9), and the primary task-function vectors $\mathbf{e}_p^{[1]}$ and $\mathbf{e}_p^{[2]}$, with dimension $p_1 \times 1$ and $p_2 \times 1$ respectively. $\mathbf{e}_p^{[1]}$ is the set of the p_1 primary obstacle avoidance task-functions that are based on equation (16), while $\mathbf{e}_p^{[2]}$ is the set of the p_2 primary contour following task-functions that are based on equation (17). They are therefore associated with p_1 and p_2 sensory data respectively.

One way to ensure the condition above — the final global task-function, $\mathbf{e}_1$, to be zero — is to minimize a cost-function depending on task-function vectors $\mathbf{e}_1^{[1]}$ and $\mathbf{e}_1^{[2]}$. Such an admissible cost function is following :

$$\mathcal{J} \;=\; \frac{1}{2}\,\left(\mathbf{e}_1^{[1]^T}\,\mathbf{e}_1^{[1]} \;+\; \mathbf{e}_1^{[2]^T}\,\mathbf{e}_1^{[2]}\right) \tag{18}$$

A *necessary* condition for $\mathcal{J}$ to be zero is that its gradient be $\mathbf{0}$, leading to the following admissible task-function :

$$\mathbf{e}_1 \;=\; \frac{\partial \mathcal{J}}{\partial \mathbf{r}} \;=\; \left(\frac{\partial \mathbf{e}_1^{[1]}}{\partial \mathbf{r}}\right)^T \mathbf{e}_1^{[1]} \;+\; \left(\frac{\partial \mathbf{e}_1^{[2]}}{\partial \mathbf{r}}\right)^T \mathbf{e}_1^{[2]} \tag{19}$$

It is easy to show that, regarding the primary task-functions defined through (16) and (17), the Hessian of $\mathcal{J}$, $\frac{\partial^2 \mathcal{J}}{\partial \mathbf{r}^2}$, is ensured to be definite positive along the ideal trajectory $\mathbf{r}^*(t)$, solution of $(\mathbf{e}_1^{[2]}(\mathbf{r},t), \mathbf{e}_1^{[2]}(\mathbf{r},t)) = (\mathbf{0},\mathbf{0})$. This positivity of the Hessian of $\mathcal{J}$ induces the positivity of the Jacobian of the task-function $\mathbf{e}_1$, defined in equation (19), which is useful for control purposes as emphasized further in Section 6. Indeed, the Hessian of $\mathcal{J}$ is a $n \times n$ matrix an ij-th element of which is :

$$\left[\frac{\partial^2 \mathcal{J}}{\partial \mathbf{r}^2}\right]_{ij} = \sum_{k=1}^{p_1} \left(\frac{\partial \mathbf{e}_{1_k}^{[1]}}{\partial \mathbf{r}_j}\frac{\partial \mathbf{e}_{1_k}^{[1]}}{\partial \mathbf{r}_i} + \frac{\partial^2 \mathbf{e}_{1_k}^{[1]}}{\partial \mathbf{r}_j \partial \mathbf{r}_i}\mathbf{e}_{1_k}^{[1]}\right) + \sum_{k=1}^{p_2} \left(\frac{\partial \mathbf{e}_{1_k}^{[2]}}{\partial \mathbf{r}_j}\frac{\partial \mathbf{e}_{1_k}^{[2]}}{\partial \mathbf{r}_i} + \frac{\partial^2 \mathbf{e}_{1_k}^{[2]}}{\partial \mathbf{r}_j \partial \mathbf{r}_i}\mathbf{e}_{1_k}^{[2]}\right)$$

that can be put under the following form :

$$\frac{\partial^2 \mathcal{J}}{\partial \mathbf{r}^2} = \frac{\partial \mathbf{e}_1^{[1]}}{\partial \mathbf{r}}^T \frac{\partial \mathbf{e}_1^{[1]}}{\partial \mathbf{r}} + \frac{\partial \mathbf{e}_1^{[2]}}{\partial \mathbf{r}}^T \frac{\partial \mathbf{e}_1^{[2]}}{\partial \mathbf{r}} + \mathbf{\Psi}_1 \mathbf{e}_1^{[1]} + \mathbf{\Psi}_2 \mathbf{e}_1^{[2]} \tag{20}$$

This equation is clearly definite positive for $(\mathbf{e}_1^{[2]}(\mathbf{r},t), \mathbf{e}_1^{[2]}(\mathbf{r},t)) = (\mathbf{0},\mathbf{0})$. A sufficient condition for $\mathcal{J}$ to be zero whenever $\mathbf{e}_1$ is $\mathbf{0}$ is then ensured. The Hessian of $\mathcal{J}$ still remains definite positive when $(\mathbf{e}_1^{[2]}(\mathbf{r},t), \mathbf{e}_1^{[2]}(\mathbf{r},t))$ is kept close enough to $(\mathbf{0},\mathbf{0})$, in a zone that depends on $\frac{\partial \mathbf{e}_1^{[1]}}{\partial \mathbf{r}}$ and $\frac{\partial \mathbf{e}_1^{[2]}}{\partial \mathbf{r}}$ and that is difficult to quantify analytically.

5 Switching between primary task-functions

Instead of dealing with the two main tasks that can be affected to an AUV through the minimization of a cost function as above, we would like to *smoothly* switch from one task to another in order to improve the transient phases in some extreme cases such as vertical cliffs and to control the vehicle behavior during such phases. This approach is general enough to be applied to other types of sensory data servoing such as visual one [Pissard-Gibollet, 1993]. Let us assume that we want to smoothly commute, around a time value t_0, from a sensor-based task-function vector $\mathbf{e}_1^{[1]}$ with dimension $m_1 \times 1$ to another sensor-based task-function vector $\mathbf{e}_1^{[2]}$ with dimension $m_2 \times 1$. Since the spaces spanned by the corresponding interaction screws $\mathbf{L}_i$ with dimension $n \times p_i$, $i = 1, 2$, may be different, two situations can occur :

5.1 $R(\mathbf{L}_1) = R(\mathbf{L}_2)$

In this case, there exists a matrix $\mathbf{W}$ such that $R(\mathbf{W}^T) = R(\mathbf{L}_1) = R(\mathbf{L}_2) = \mathcal{S}$ with $m_1 = m_2 = m$. Defining the matrices $\mathbf{C}_1$ and $\mathbf{C}_2$ as $\mathbf{C}_1 = \mathbf{W}\mathbf{L}_1$ and $\mathbf{C}_2 = \mathbf{W}\mathbf{L}_2$ respectively, a new merging primary task-function is :

$$\mathbf{e}_1 \;=\; \lambda(u)\; \mathbf{e}_1^{[1]} \;+\; (1 \;-\; \lambda(u))\; \mathbf{e}_1^{[2]} \tag{21}$$

with

- $\mathbf{e}_1^{[1]}$ and $\mathbf{e}_1^{[2]}$ defined as in (8) using $\mathbf{C}_1$ and $\mathbf{C}_2$ as defined above, and the basic primary task-function vectors $\mathbf{e}_p^{[1]}$ and $\mathbf{e}_p^{[2]}$ as in (16) and (17) with dimension $p_1 \times 1$ and $p_2 \times 1$ respectively.
- $\lambda(u)$ is a $\mathcal{C}^2$ scalar function decreasing on [0,τ] such that $\lambda(0) = 1$ and $\lambda(\tau) = 0$. τ is the time constant for the complete commutation between $\mathbf{e}_1^{[1]}$ and $\mathbf{e}_1^{[2]}$, and $u = t - t_0 + \tau/2$. It is worth noticing that this choice

of u is purely arbitrary as its only goal is to let $\lambda(t - \tau/2)$ be an odd function. A simple example of $\lambda(u)$ is :

$$\lambda(u) = -\frac{1}{\tau^3}u^3 + 1 \tag{22}$$

The Jacobian matrix of $\mathbf{e}_1$ is :

$$\mathbf{J}_1 \;=\; \frac{\partial \mathbf{e}_1}{\partial \mathbf{r}} \;=\; \mathbf{W}\;[\lambda\;\mathbf{L}_1\;\mathbf{L}_1^T \;+\; (1-\lambda)\;\mathbf{L}_2\;\mathbf{L}_2^T] \tag{23}$$

It can be shown that it exists a matrix $\mathbf{D}$ with dimension $p_2 \times p_1$ and rank $m \leq p_i, i = 1, 2$, such that $\mathbf{L}_1 = \mathbf{L}_2\mathbf{D}$. Indeed, consider $\{\mathbf{f}_1, \ldots \mathbf{f}_m\}$, a basis of $\mathcal{S}$. Denote $\mathbf{L}_1^k$ and $\mathbf{L}_2^k$, the k-th columns of $\mathbf{L}_1$ and $\mathbf{L}_2$ respectively. We can write :

$$\begin{aligned}
\mathbf{L}_1^k &= \sum_{j=1}^{m} \lambda_{kj}^{[1]}\mathbf{f}_j, \quad k = 1..p_1 \\
\mathbf{L}_2^k &= \sum_{j=1}^{m} \lambda_{kj}^{[2]}\mathbf{f}_j, \quad k = 1..p_2
\end{aligned}$$

Let us define the $n \times m$ matrix $\mathbf{E} = [\mathbf{f}_1, \ldots \mathbf{f}_m]$, and $\mathbf{\Lambda}_i$ the $m \times p_i$ matrices of rank m such that :

$$\mathbf{\Lambda}_i = \begin{pmatrix} \lambda_{11}^{[i]} & \cdots & \lambda_{p_i 1}^{[i]} \\ \vdots & & \vdots \\ \lambda_{1m}^{[i]} & \cdots & \lambda_{p_i m}^{[i]} \end{pmatrix}, \quad i = 1, 2$$

This yields that : $\mathbf{L}_i = \mathbf{E}\Lambda_i$.

We search for a matrix $\mathbf{D}$ such that $\mathbf{L}_1 = \mathbf{L}_2\mathbf{D}$, therefore such that $\mathbf{E}\mathbf{\Lambda}_1 = \mathbf{E}\mathbf{\Lambda}_2\mathbf{D}$. This is true in particular if $\mathbf{\Lambda}_1 = \mathbf{\Lambda}_2\mathbf{D}$, leading to :

$$\mathbf{D} = \mathbf{\Lambda}_2^+\mathbf{\Lambda}_1 = \mathbf{\Lambda}_2^T(\mathbf{\Lambda}_2\mathbf{\Lambda}_2^T)^{-1}\mathbf{\Lambda}_1$$

As $\mathbf{\Lambda}_i$ is of dimension $m \times p_i$, and $rank(\mathbf{\Lambda}_i) = m$, we then have found a $p_2 \times p_1$ matrix of rank m that solves our problem.

It then follows that :

$$\mathbf{J}_1 \;=\; \mathbf{W}\;\mathbf{L}_2\;\mathbf{A}'\;\mathbf{L}_2^T$$

with

$$\mathbf{A}' \;=\; \lambda\;\mathbf{D}\;\mathbf{D}^T \;+\; (1-\lambda)\;\mathbf{I}_{p_2}$$

Since $\mathbf{D}\mathbf{D}^T$ is a positive symmetrical matrix, all its eigenvalues μ_i are positive. Then a sufficient condition for the positivity of $\mathbf{A}'$ is :

$$\lambda\;\mu_{min} \;+\; (1-\lambda) \;>\; 0$$

$\mathbf{A}'$ is a symmetrical definite positive matrix for $\lambda < 1$, and a positive symmetrical one when $\lambda = 1$. From a convenient choice of $\mathbf{W}$, this positivity of $\mathbf{A}'$ induces the positivity of $\mathbf{J}_1$, which is useful for control purposes as emphasized further in Section 6.

Remark When $p_1 = p_2 = m$, it is easy to show that the merging task can be done directly on the basic primary task-function vectors, i.e. on $\mathbf{e}_p^{[i]}$, $i = 1..2$.

5.2 $R(\mathbf{L}_1) \neq R(\mathbf{L}_2)$

The first possible case is when $m_1 = m_2 = m$. There exist two matrices $\mathbf{W}_i$, $i = 1..2$, such that $\mathbf{C}_i = \mathbf{W}_i \mathbf{L}_i$. Then, a possible merging primary task-function is :

$$\mathbf{e}_1 \;=\; \lambda(u)\; \mathbf{W}_1\; \mathbf{L}_1\; \mathbf{e}_p^{[1]} \;+\; (1-\lambda(u))\; \mathbf{W}_2\; \mathbf{L}_2\; \mathbf{e}_p^{[2]} \tag{24}$$

It is worth noting that there is no reason for a simultaneous achievement of both elementary primary tasks $\mathbf{e}_p^{[1]}$ and $\mathbf{e}_p^{[2]}$ since $R(\mathbf{L}_1) \neq R(\mathbf{L}_2)$. Moreover, the Jacobian of $\mathbf{e}_1$ does not exhibit any particular global property.

In the second case, where $m_1 \neq m_2$, it is no longer possible to keep working on $\mathbf{e}_1$ as above (24). Let us then define :

$$\mathbf{e} \;=\; \lambda(u)\; \mathbf{e}^{[1]} \;+\; (1-\lambda(u))\; \mathbf{e}^{[2]} \tag{25}$$

where $\mathbf{e}^{[1]}$ and $\mathbf{e}^{[2]}$ are given now by :

$$\mathbf{e}^{[i]} \;=\; \mathbf{W}_i^{+}\; \mathbf{W}_i\; \mathbf{L}_i\; \mathbf{e}_p^{[i]} \;+\; \alpha_i\; \left(\mathbf{I}_n - \mathbf{W}_i^{+}\mathbf{W}_i\right)\; \mathbf{g}_{s_i} \tag{26}$$

Then follows the Jacobian matrix :

$$\mathbf{J} \;=\; \lambda(u)\; \frac{\partial \mathbf{e}^{[1]}}{\partial \mathbf{r}} \;+\; (1-\lambda(u))\; \frac{\partial \mathbf{e}^{[2]}}{\partial \mathbf{r}} \tag{27}$$

With a convenient choice of $\mathbf{e}^{[i]}$, every $\frac{\partial \mathbf{e}^{[i]}}{\partial \mathbf{r}}$ is definite positive along its own ideal trajectory. Nevertheless, the ideal trajectory for $\mathbf{e}$ may be different during the transient phase, and no conclusion can be done concerning the definite positiveness of the Jacobian.

A last possibility, which is generally used, consists in setting an intermediate stopping phase at a given position before commuting to another task. The two tasks do not overlap and the switching is more regular :

Phase 1 Let us define :

$$\tilde{\mathbf{e}} \;=\; \lambda(u)\; \mathbf{e}^{[1]} \;+\; (1-\lambda(u))\; \boldsymbol{\xi}^{*} \tag{28}$$

where

$$\boldsymbol{\xi}^{*} \;=\; \begin{bmatrix} \mathbf{x} - \mathbf{x}^{*} \\ \mathbf{O}_e(\mathbf{r}, \mathbf{r}^{*}) \end{bmatrix} \tag{29}$$

with $\mathbf{x}^{*}$ the position and $\mathbf{O}_e(., \mathbf{r}^{*})$ a parameterization of the current attitude of the vehicle at $u = 0$.

It appears that :

$$\frac{\partial \tilde{\mathbf{e}}}{\partial \mathbf{r}} \;=\; \lambda(u)\,\frac{\partial \mathbf{e}^{[1]}}{\partial \mathbf{r}} \;+\; (1-\lambda(u))\,\begin{bmatrix} \mathbf{I}_3 & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Upsilon} \end{bmatrix} \tag{30}$$

As before, $\frac{\partial \mathbf{e}^{[1]}}{\partial \mathbf{r}}$ is definite positive along its ideal trajectory, with $\mathbf{r}^* = \mathbf{r}(u=0)$ belonging to this ideal trajectory when the initial value of the task, $\tilde{\mathbf{e}}(u=0)$, is small enough. $\begin{bmatrix} \mathbf{I}_3 & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Upsilon} \end{bmatrix}$ is definite positive if the attitude parameterization is well chosen. The resulting situation is then better than previously.

Phase 2 Once the equilibrium $\tilde{\mathbf{e}}(u=0)$ is achieved, the second task $\tilde{\mathbf{e}}'$ is activated :

$$\tilde{\mathbf{e}}' \;=\; \lambda(t-t_0)\;\boldsymbol{\xi}^* \;+\; (1-\lambda(t-t_0))\;\mathbf{e}^{[2]} \tag{31}$$

Note however that there is no reason for $\mathbf{e}^{[2]}$ to vanish at $t=t_0$.

As far as we are concerned with the particular case of the AUVs, the switching appropriate task-function will be defined as studied in Subsection 5.2, since there is no reason for having equality between the configuration spaces generated by the obstacle avoidance and the surface following tasks respectively. Moreover, due to motion requirements induced by some load sensors, the vehicle may not be able to stop at an intermediate configuration, since it has to move at a constant nominal axial speed, u_0, even when facing vertical cliffs. In that latter case, there is no obvious way to define $\boldsymbol{\xi}^*$ in (28). Therefore, we have to use a degraded switching task based on equation (25), which nevertheless will prove to be adequate.

6 A sensor-based control law for AUVs

Modelling of the dynamics of underwater vehicles [Fossen, 1994] leads to the following general state equation :

$$\mathbf{M}\,\dot{\mathbf{w}} \;=\; \mathbf{G}(\mathbf{w})\;\mathbf{w} \;+\; \mathbf{D}(\mathbf{w})\;\mathbf{w} \;+\; \boldsymbol{\Gamma}_g \;+\; \boldsymbol{\Gamma}_p \;+\; \boldsymbol{\Gamma} \tag{32}$$

where

- $\mathbf{M}$, the inertia matrix, is a positive definite symmetric matrix ;
- $\mathbf{w}$ is the vehicle velocity vector expressed in its local frame ;
- $\mathbf{G}(\mathbf{w})\mathbf{w}$ is the vector of Coriolis and centrifugal forces, with $\mathbf{G}(\mathbf{w})$ a skew-symmetric matrix ;

- $\mathbf{D}(\mathbf{w})\mathbf{w}$ is the vector of hydrodynamic forces, except the buoyancy, with $\mathbf{D}(\mathbf{w})$ a negative definite matrix ;
- $\mathbf{\Gamma}_g$ includes the gravity and buoyancy forces and moments ;
- $\mathbf{\Gamma}_p$ refers the disturbance effects on the dynamics including currents ;
- $\mathbf{\Gamma}$ is the vector of forces and moments generated by the vehicle actuators.

From the first derivative of a hybrid task-function $\mathbf{e}$ with respect to time, and reminding that $\mathbf{e}$ is both a function of time and of $\mathbf{r}$, we obtain :

$$\dot{\mathbf{e}}(\mathbf{r},t) \;=\; \frac{\partial \mathbf{e}}{\partial \mathbf{r}}(\mathbf{r},t)\;\mathbf{w} \;+\; \frac{\partial \mathbf{e}}{\partial t}(\mathbf{r},t) \tag{33}$$

In order that $\mathbf{e}$ behaves like a first order decoupled system exponentially decreasing, the desired evolution of $\mathbf{e}$ should have the form :

$$\dot{\mathbf{e}} \;=\; -k_e\;\mathbf{e} \text{ with } k_e \;>\; 0 \tag{34}$$

Combining (33) and (34) then leads to the following ideal velocity vector $\mathbf{w}_d$ that has to be achieved on the vehicle through the vehicle dynamics (32) :

$$\mathbf{w}_d \;=\; \left(\widehat{\frac{\partial \mathbf{e}}{\partial \mathbf{r}}}\right)^{-1}\left(-\;k_e\;\mathbf{e}\;-\;\widehat{\frac{\partial \mathbf{e}}{\partial t}}\right) \tag{35}$$

where $\widehat{x}$ represents the estimate of the expression x. An estimation procedure of the term $\widehat{\frac{\partial \mathbf{e}}{\partial t}}(\mathbf{r},t)$ is presented in [Santos and Chaumette, 1992].

Rewriting the vehicle dynamics (32) as :

$$\left\{\begin{array}{rcl} \mathbf{\Gamma} & = & \mathbf{M}\,\mathbf{\Gamma}_a \;-\; \mathbf{G}(\mathbf{w})\;\mathbf{w} \;-\; \mathbf{\Gamma}_g(\mathbf{w}) \;-\; \mathbf{\Gamma}_p(\mathbf{w}) \\ \dot{\mathbf{w}} & = & \mathbf{\Gamma}_a \end{array}\right. \tag{36}$$

directly shows the linear decoupling process. Indeed, when $\mathbf{\Gamma}$ can be exactly applied to the vehicle through the vehicle actuators, thus leading to a holonomic vehicle, several robust control approaches, either linear or nonlinear, can be investigated, even used depending on the vehicle actuation and level of identification and, of course, its environment and motion constraints, :

- Feedback Linearization,
- robust LQG Control [Santos and Bitmead, 1995],
- H_∞ [Fryxell et al., 1994],
- Nonlinear Feedback [de Wit et al., 1993, Perrier et al., 1994],
- Adaptive Control [Fossen and Fjellstad, 1994]

- Sliding Control [Slotine and Li, 1991, Cristi et al., 1990],
- Fuzzy Control [Kato, 1994],
- and many others...

Considering the Feedback Linearization, it can be easily shown that taking the pseudo-control variable $\mathbf{\Gamma}_a$ as :

$$\mathbf{\Gamma}_a \;=\; \dot{\mathbf{w}}_d \;-\; k_a\,(\mathbf{w} \;-\; \mathbf{w}_d) \tag{37}$$

suffices to ensure a robust stability behavior of the vehicle under two types of conditions :

(i) conditions on the minimal values of the gains k_a, k_e,

(ii) conditions on the used models, either concerning the dynamics of the vehicle, or related to the task-function [Samson et al., 1990] :

$$\left(\frac{\partial \mathbf{e}}{\partial \mathbf{r}}\right)\left(\widehat{\frac{\partial \mathbf{e}}{\partial \mathbf{r}}}\right)^{-1} \;>\; \mathbf{0}$$

This latter equation shows that, if it can be ensured that the true Jacobian matrix of $\mathbf{e}$ is positive, even when not exactly known, a very simple model may be used in the control, the identity matrix $\mathbf{I}_n$ being the extremal case.

Remark In the case of a non-zero steady-state error on $\mathbf{e}$, or when $\widehat{\frac{\partial \mathbf{e}}{\partial t}}$ cannot be estimated, an upper bounded integral feedback of $\mathbf{e}$ may be included in the equation (35) determining $\mathbf{w}_d$.

7 Simulation results

A simulator of a fully-actuated underwater vehicle has been built around the SIMPARC kernel software [Borrelly and Astraudo, 1992], using a complete dynamic model of a torpedo-shaped AUV with a 7-meter length and 1-meter diameter. The vehicle is considered to have two vertical screw propellers, two lateral ones, and an axial one. These actuators are assumed not to be saturated.

The SIMPARC software consists in a set of C++ classes designed to build simulators for multiprocessor robot controllers . It is built on an event driven simulator for program execution simulation, and on a numerical integration package for robots behavior simulation (see Figure 7 for a general organization of a SIMPARC scheme). It allows the hardware description of the target architecture in terms of communicating devices. This software allows the user

to define both its own robot models, and furthermore the interaction between exteroceptive sensors and the environment. It is worth noticing that the exteroceptive sensors and the environment can be separately modelled. SIMPARC actually runs user programs on the simulated architecture. It is mainly concerned with timing aspects of multiprocessor applications and takes into account the sharing of communication resources and processor interrupts.

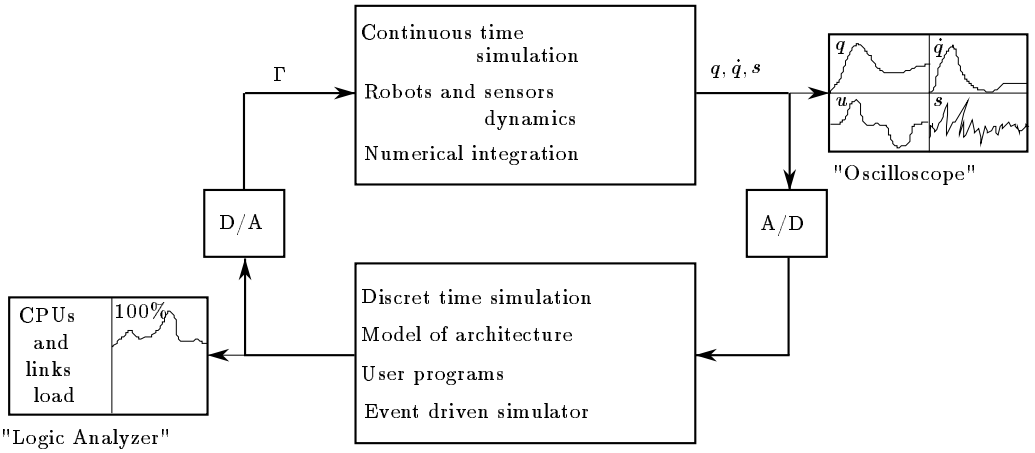


Figure 7: Schematic overview of the implementation of a dynamic system on SIMPARC software

For simplicity and without any loss of generality, the motion of the vehicle is restricted to the vertical motion plane. We assume that the vehicle has a vertical symmetry plane, which is generally the case, leading to no coupling effects with the horizontal plane.

We would like to achieve the following of a profile-varying and *a priori* unknown sea bottom at a desired reference distance $y_d \pm \Delta y_d$ with a nominal constant axial speed u_0 within the range 1–3m/sec, whatever the initial vehicle configuration and the bottom profile are. In the vertical plane, the vehicle has 3 degrees of freedom : translation along Ox, translation along Oz and rotation around Oy. Since the translational Ox degree of freedom is assumed to be already constrained by the desired axial velocity, there are two others left that can be constrained by the sea-bottom following task, leading to $n = 2$. To achieve this bottom following task, both contour following and obstacle avoidance tasks are required. A user-defined bottom profile is defined with positive or negative slope breaks up to $\pm$90deg. It is worth noting that the vehicle is intended to remain parallel to the bottom profile. Two load sensors are considered for simulations, as they are expected to be the most critical ones in operation if we refer to the resulting required altitude of the vehicle with respect to the bottom [Masson et al., 1994] :

the TOBI sidescan sonar It is a 30kHz sidescan sonar. The required altitude is 400±50m.

the camera The required altitude should be within the range 5–25m with an accuracy of ±5m or better.

We will consider the following altitude constraints $y_d \pm \Delta y_d$ evaluated at the bottom of the vehicle :

- 400m±50m,
- 25±5m,
- and 5m±as-few-as-possible

at respective nominal constant axial velocity u_0 of 1m/sec and 3m/sec. Then the resulting altitude of the origin of the vehicle local frame is $r_v + y_d \pm \Delta y_d$, where r_v is the radius of the cylindrical body of the vehicle. These constraints are required when slope values are within ±45degrees. For each constraint, smooth profiles, corners and edges are considered, and the results that are obtained when determining task-functions using concepts defined in Section 4 and Section 5, are compared. For the law-level control, the Feedback Linearization approach (35)–(37) is considered.

7.1 Smooth profiles

As emphasized in Subsection 3.2, we consider the bottom profile to be smooth, or with no break, when $y_d < r_{min}$, with r_{min} the minimal value of the radius of any admissible circle tangenting the reference trajectory curve $(\mathcal{C}_t)$ at two or more points, and the interior of which does not contain any point of the curve.

Since no obstacle avoidance task is required, only the contour following one has to be considered. The virtual linkage that is convenient for this task is the line/plane linkage presented in Subsection 3.2. Two sensory data δ_1 and δ_2 are required at least. These data are assumed to be obtained from sensors $(S1)$ and $(S2)$ with constant angle orientations in the vehicle local frame of β_1 and β_2 respectively (see Figure 8). The reference sensory data should clearly be set to :

$$\delta_i^* = \frac{y_d}{\cos(\beta_i)}, \quad i = 1, 2, \quad |\beta_i| < \frac{\pi}{2}, \quad \delta_i^* < \delta_{max} \tag{38}$$

The resulting primary task-function vector $\mathbf{e}_p$, based on equation (17) and of dimension 2×2, is :

$$\mathbf{e}_p = k_2 \begin{pmatrix} \delta_1 - \frac{y_d}{\cos(\beta_1)} \\ \delta_2 - \frac{y_d}{\cos(\beta_2)} \end{pmatrix}, \quad k_2 > 0 \tag{39}$$

The interaction screw $\mathbf{L}$ in (2) is defined using the approximation (15), and is expressed in the vehicle local frame at its origin O. The final task-function vector $\mathbf{e}$ to regulate to zero is then simply defined by applying directly equations (5)-(9). As $n = m = p = 2$ in this case, the combination matrix is not required, so $\mathbf{C}$ can be chosen as the identity matrix $\mathbf{I}_2$. So is $\mathbf{W}$. No secundary task is admissible. Therefore, the final task-function to regulate to $\boldsymbol{a}$ is : $\mathbf{e} = \mathbf{e_p}$, with $\mathbf{e}_p$ defined as in (39).

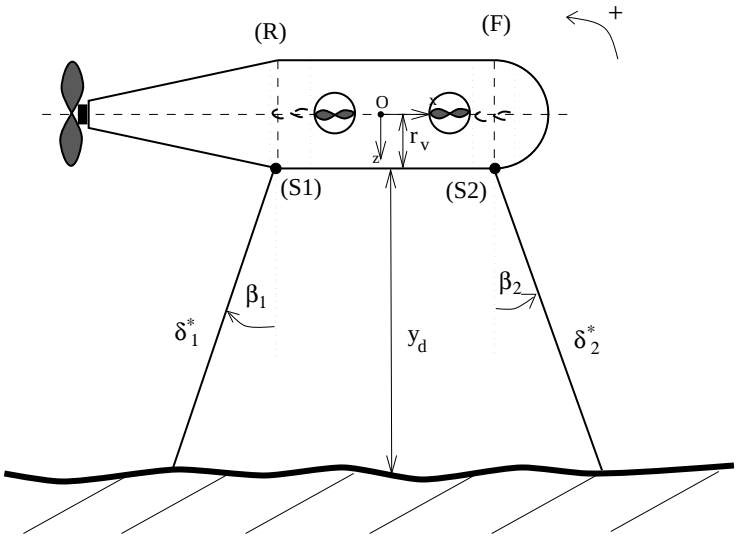


Figure 8: Sensors configuration for a smooth profile following

In the simulation depicted on Figure 9, the desired altitude is y_d =5m. The nominal axial velocity u_0 is the maximal admissible one, 3m/sec. This is the worst case for bottom following. For a slight predictive perception, β_1 and β_2 are set to -10deg and 10deg respectively (could be set to 0deg as well for lower velocities such as 1m/sec with no significant change). The bottom is composed of successive lines and arcs of circle the ray of curvature of which is strictly greater than $y_d + \Delta y_d$. The vehicle is initiated at an altitude of $y(0)$ =50m and a pitch angle of $\theta(0)$ =20deg. The controller parameters have been set to : $k_2 = 1$, $k_a = 0.15$ and $k_e = 0.3$. As can be seen on Figure 9, the convergence and the smooth surface following are well done (chart [a]), and the altitude error (chart [b]) in steady-state phases remains less than 0.5m. During the phases with curves, this error increases up to 3m, and further simulations show that it significantly decreases for lower axial velocities (2m/sec or 1m/sec) and for higher desired altitude. Further simulations show that the error can conveniently be maintained up to 5m for y_d =25m and up to 20m for y_d =400m for curves with ray of curvature of greater than the y_d value. This bounding error is mainly function of :

- the sensors orientation and their points of contact (or impacts) on the ground (recall that the bottom is viewed from the vehicle as a line between the points of contact),
- the curvature of the bottom profile at the vertical projection of the vehicle origin on the ground,
- the vehicle's advancement velocity and desired reference altitude,
- the vehicle's actuation dynamics (e.g. saturation),
- and, of course, the values of the controller parameters (k_2, k_a and k_e) that are used.

The complexity and the large extend of the parameters cited above do not make it possible to give a quantitative analysis of the bounding error Δy_{max} in the general case of any vehicle dynamics and any law level controller. Only the well-known *trial-and-error* procedure can give the answer.

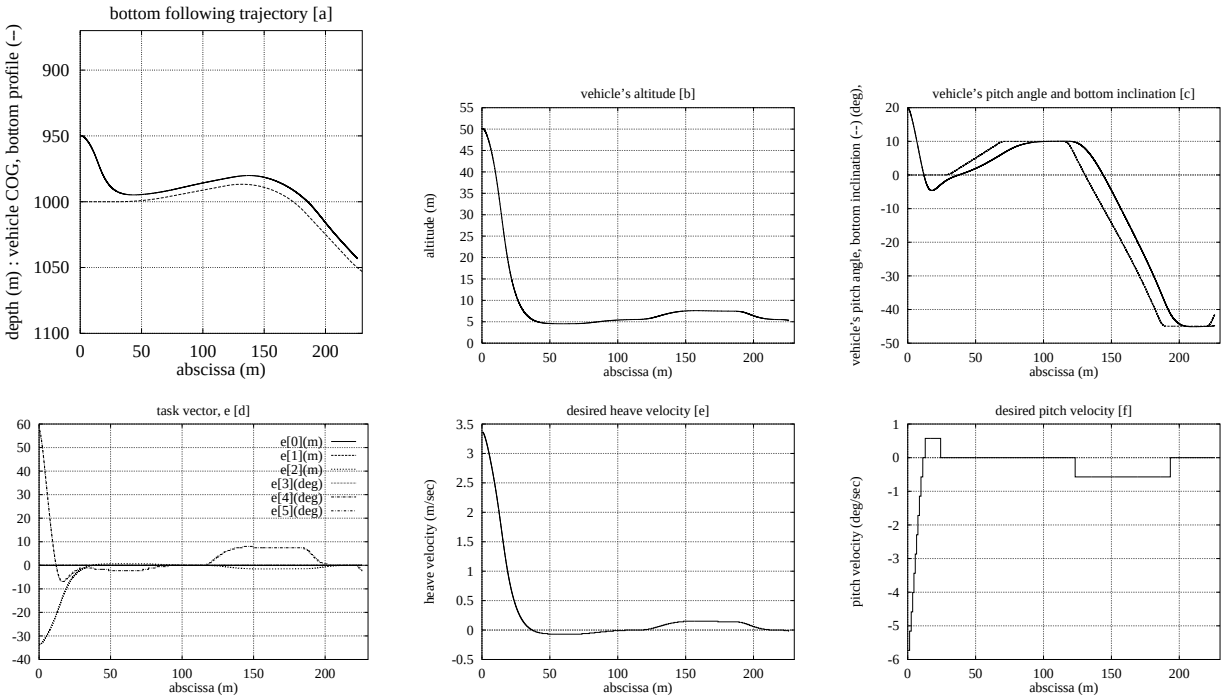


Figure 9: Bottom following of a smooth profile : y_d =5m, u_0 =3m/sec, $\beta_1 = -10deg$ and β_2 =10deg, $y(0)$ =50m and $\theta(0)$ =20deg, $k_2 = 1$, $k_a = 0.15$ and $k_e = 0.3$

7.2 Edges

The break is in the clockwise direction, with $\Delta\alpha < 0$. We will consider the worst case where $r_{min} = 0$ (see Figure 10).

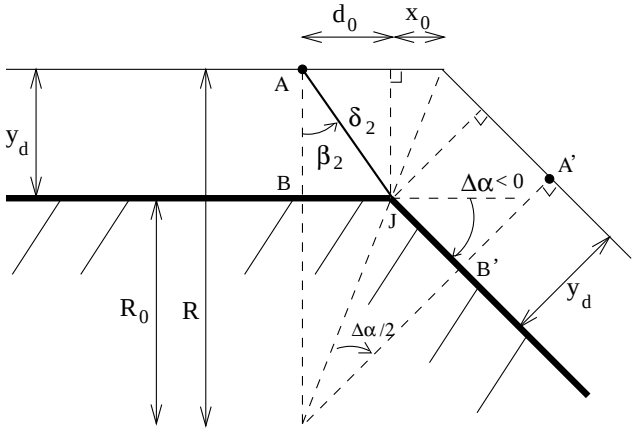


Figure 10: Example of an edge

As an edge cannot be considered as an obstacle (no collision expected), only the contour following task has to be considered as in Subsection 7.1 above. Again the virtual linkage that is convenient for this task is the line/plane linkage, with two required sensory data at least. Let consider they are obtained from sensors $(S1)$ and $(S2)$ with respective orientation β_1 and β_2. For the edge to be detected at least before the path junction, the condition is :

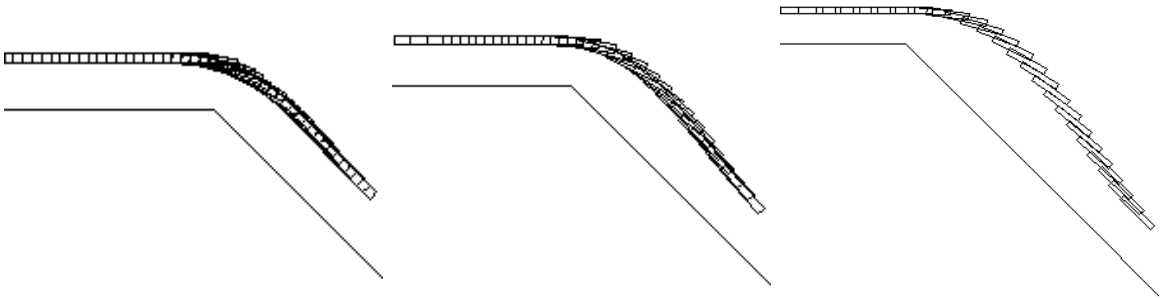
$$\beta_2 \geq \Delta\alpha/2 \tag{40}$$

with $-45\text{deg} < \Delta\alpha < 0$ from our operational specifications.

The resulting task-function vector $\mathbf{e}$ may be defined just as in (39) presented in the previous Subsection 7.1. The transient phase when passing along the edge is difficult to quantify. Indeed, as depicted on Figures 11(a)–(c) and Figures 12(a)–(b) dealing with abrupt edges with $r_{min} = 0$, it mainly depends on the slope break value, the desired vehicle's advancement velocity reference altitude, the reference altitude, the sensors orientation, and the values of the controller parameters. It is worth noticing that the higher the ray of curvature of the ideal trajectory and the r_{min} value of the bottom profile are, the better the transient is (see Figure 13).

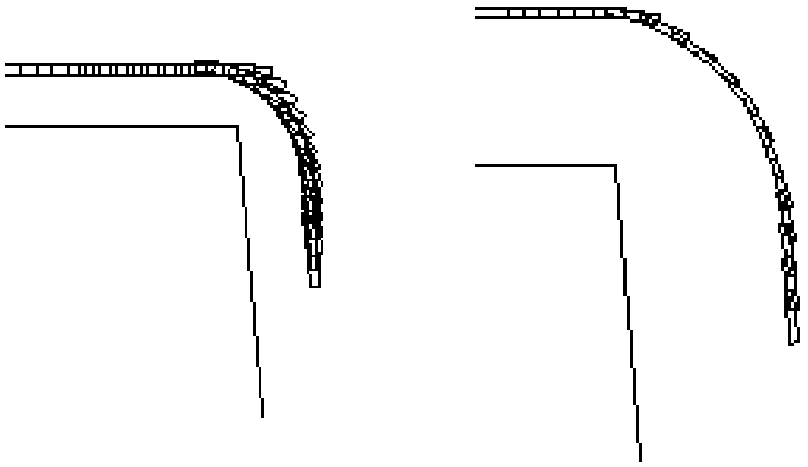
A more precise approach is to switch, smoothly whenever possible, from the contour following of the part of the profile that is smooth to a virtual path following the ray of curvature of which can be defined as :

$$R = y_d \left(\frac{\tan\beta_2}{\tan(\Delta\alpha/2)} - 1 \right) \quad < 0 \tag{41}$$



(a) u_0 =1m/s (b) u_0 =2m/s (c) u_0 =3m/s

Figure 11: Simulation results at various advancement velocities for a reference altitude of 5m when passing an abrupt edge ($r_{min} = 0$) with a break slope value of 45deg. $\beta_1 = \beta_2$ =0deg, $k_2 = 1$, $k_a = 0.15$ and $k_e = 0.3$. No switching task.



(a) u_0 =1m/s, y_d =5m (b) u_0 =3m/s, y_d =20m

Figure 12: Simulation results at various advancement velocities and various reference altitudes when passing an abrupt edge ($r_{min} = 0$) with a break slope value of 85deg. β_1 =0deg, β_2 =10deg, $k_2 = 1$, $k_a = 0.15$ and $k_e = 0.3$. No switching task.

leading to a desired pitch velocity $q_d = u_0/R$ to be applied from A to A' on Figure 10. This is equivalent to consider a virtual smooth bottom profile between point B and B' the ray of curvature of which is :

$$R_0 = y_d \frac{\tan \beta_2}{\tan(\Delta\alpha/2)} \quad < 0 \tag{42}$$

This can be done using the concept developed in Section 5. The aim is to switch from the task of following the smooth part of the bottom to the

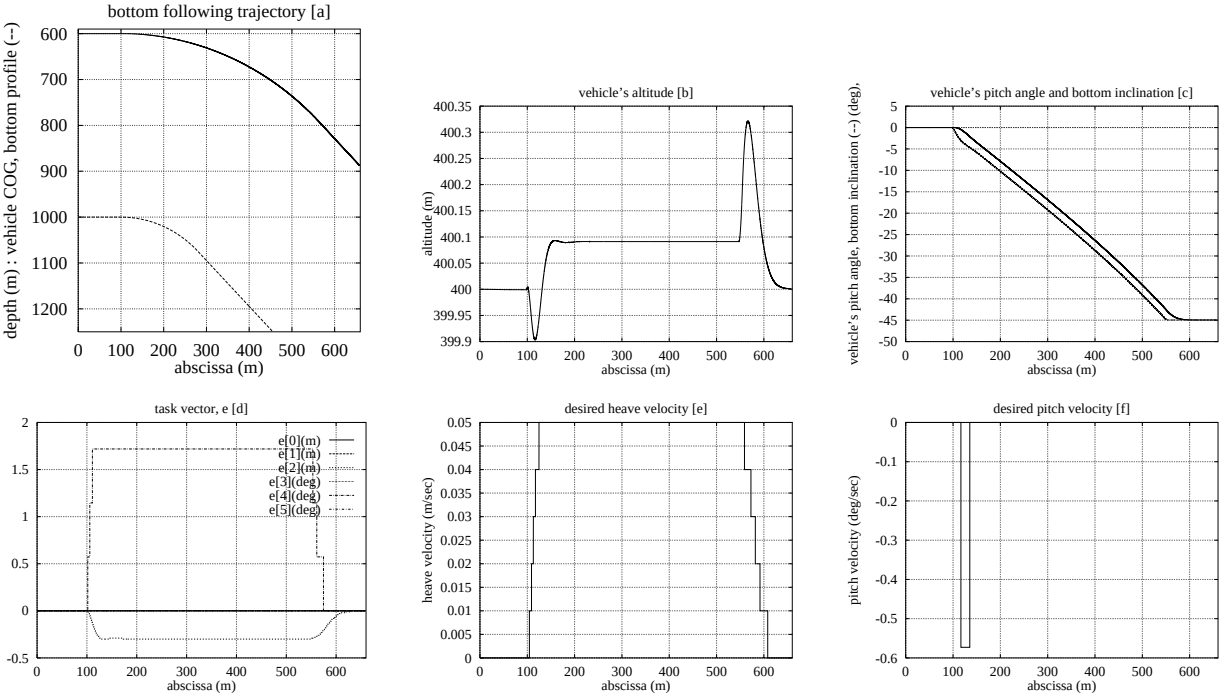


Figure 13: Bottom following of an edge with $y_d > r_{min} > 0$: y_d =400m, u_0 =3m/sec, $\beta_1 = \beta_2$ =0deg, $y(0)$ =400m and $\theta(0)$ =0deg, $k_2 = 1$, $k_a = 0.15$ and $k_e = 0.3$

following of the junction trajectory to the next smooth part of the bottom, and finally switch to the smooth bottom part following task after the edge is passed. Let consider the first task vector $\mathbf{e}^{[1]}$ achieving the following of the smooth bottom part as follows (before and after the edge) :

$$\mathbf{e}^{[1]} = k_2 \begin{pmatrix} \delta_1 - \frac{y_d}{\cos(\beta_1)} \\ \delta_2 - \frac{y_d}{\cos(\beta_2)} \end{pmatrix}, \quad k_2 > 0 \tag{43}$$

The task vector achieving the junction path following can be defined following :

$$\mathbf{e}^{[2]} = k \begin{pmatrix} z_C - z^*(t) \\ \theta_C - \theta^*(t) \end{pmatrix}, \quad k > 0 \tag{44}$$

where z_C and θ_C represent the current depth and pitch angle of the vehicle respectively. $z^*(t)$ and $\theta^*(t)$ are the reference depth and pitch angle respectively, that can be defined as :

$$\begin{aligned} z^*(t) &= z_A - R(1 - \cos(\theta^*)) \\ \theta^*(t) &= \theta_A + \frac{u_0}{R} t \end{aligned}$$

where z_A and θ_A are the vehicle situation parameterization at the point A depicted on Figure 10, u_0 is the vehicle's nominal advancement velocity, and t

stands as the time variable within $[0, T]$ with T defined as :

$$T = \frac{|R|}{u_0}\Delta\alpha$$

Obviously, $R(\mathbf{L}_1) \neq R(\mathbf{L}_2)$ with $m_1 = m_2 = m$. The final task-function vector to regulate to zero can then be written :

$$\mathbf{e} \;=\; \lambda(u)\;\mathbf{e}^{[1]} \;+\; (1-\lambda(u))\;\mathbf{e}^{[2]} \tag{45}$$

where $\lambda(u)$ is a $\mathcal{C}^2$ scalar function decreasing on $[0,\tau]$ such that $\lambda(0) = 1$ and $\lambda(\tau) = 0$ as in (22), with τ the time constant for the complete commutation between $\mathbf{e}^{[1]}$ and $\mathbf{e}^{[2]}$. As already emphasized in Subsection 5.2, even if every $\frac{\partial \mathbf{e}^{[i]}}{\partial \mathbf{r}}$ is definite positive along its own ideal trajectory, the Jacobian of $\mathbf{e}$ does not exhibit any particular global property during the transient phase. Nevertheless detailed simulations confirm the fact that the resulting trajectories are quite smooth, and that they achieve the operational requirements for any load sensor that is considered.

One should however ensure that the resulting ideal junction trajectory during the edge passing is located inside the "tube" defined by the reference altitude range constraint, $y_d \pm \Delta y_d$. For that to hold, a constraint is put on β_2 through the following inequality :

$$|R|\left(1 - \cos(\frac{\Delta\alpha}{2})\right) < \Delta y_d \tag{46}$$

Remark The edge detection can be simply done by applying a jump detection procedure (example of Cumulative Sum test) on the sensory data δ_2. Considering the estimation of $\Delta\alpha$, it can be done through geometrical reasonings using two front sensors [Santos et al., 1995].

7.3 Corners

The break is in the counterclockwise direction, with $\Delta\alpha > 0$. We will consider the worst case where $r_{min} = 0$ (see Figure 14).

Now both smooth contour following and obstacle avoidance should be taken into account. The convenient virtual linkages are :

- the line/plane linkage, with two required sensory data at least.
- the point/plane linkage, with one front sensory data at least.

Then the minimal number of sensory data involved is greater or equal to three, with two dedicated to the contour following primary task at least , and one front dedicated to the obstacle avoidance task. Let consider the minimal configuration composed of three sensors, and let call the involved sensory data δ_1,

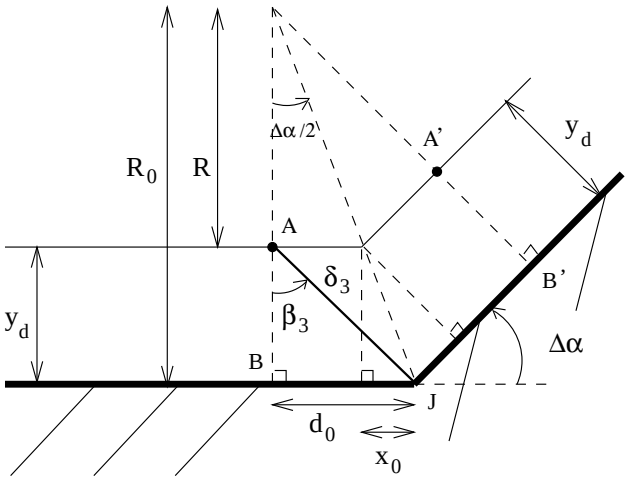


Figure 14: Example of a corner

δ_2, and δ_3 respectively. Let consider they are obtained from sensors $(S1)$, $(S2)$, and $(S3)$ with respective orientation β_1, β_2, and β_3. For the corner to be detected before the path junction, the condition is :

$$\beta_3 \geq \Delta\alpha/2 \tag{47}$$

with $0 < \Delta\alpha < 45\text{deg}$ from our operational specifications.

Let define the obstacle avoidance and contour following primary task-function vectors, $\mathbf{e}_1^{[1]}$ and $\mathbf{e}_1^{[2]}$ of dimension $p_1 \times 1$ and $p_2 \times 1$ respectively. With a three-sensor configuration as defined above, we have $p_1 = 1$ and $p_2 = 2$. The initial primary task-function vectors can be expressed as :

$$\mathbf{e}_p^{[1]} = \begin{cases} -\,k_1 \left(\dfrac{\delta_m}{\delta_3} - 1 \right), & \text{when } \delta_3 \leq \delta_m \\ 0, & \text{when } \delta_3 > \delta_m \end{cases} \tag{48}$$

$$\mathbf{e}_p^{[2]} = k_2 \begin{pmatrix} \delta_1 - \frac{y_d}{\cos(\beta_1)} \\ \delta_2 - \frac{y_d}{\cos(\beta_2)} \end{pmatrix}, \quad k_2 > 0 \tag{49}$$

The corresponding interaction screws, $\mathbf{L}_1$ and $\mathbf{L}_2$, in (2) are defined using the approximation (15), and are both expressed in the vehicle local frame at its origin O. The resulting task-function vector $\mathbf{e}$ to regulate to $\mathbf{0}$ can now be defined using one of the two approaches that have been presented in Section 4 and Section 5.

When using the minimization of a cost function concept, $\mathbf{e}$ is defined as in (19) with the cost function defined as in (18). The expressions of involved $\mathbf{e}^{[1]}$ and $\mathbf{e}^{[2]}$ can be defined as following :

$$\begin{aligned} \mathbf{e}^{[1]} &= \mathbf{W}_1^+ \mathbf{W}_1 \mathbf{L}_1 \mathbf{e}_p^{[1]} + \alpha_1 \left(\mathbf{I}_2 - \mathbf{W}_1^+ \mathbf{W}_1 \right) \mathbf{g}_{s_1} \\ \mathbf{e}^{[2]} &= \mathbf{e}_p^{[2]} \end{aligned}$$

A possible choice of $\mathbf{W}_1$ is $\mathbf{W}_1 = \mathbf{L}_1^T$ and $\mathbf{g}_{s_1}$ is the gradient of a secundary task h_s to be minimized under the constraint of $\mathbf{W}_1\mathbf{L}_1\mathbf{e}_p^{[1]} = 0$. As an example, one could like to minimize the following :

$$h_s = \frac{1}{2}\left(\delta_1 - \frac{y_d}{\cos(\beta_1)}\right)^2$$

corresponding to the task contour following achievement on sensor $(S1)$ (the rear sensor).

Figures 15(a)–(c) emphasize that the orientation of the front sensor plays a role in the behavior of the vehicle during the transient phase, as well as the reference altitude and the advancement velocity. When the orientation of the front sensor is lesser than the slope break value, the behavior is oscillatory. Indeed, a study of the variation of a front sensory data δ with respect to a positive slope break $\Delta\alpha$ (see Figure 16) yields :

$$\frac{\partial\delta}{\partial x} = \sin\beta - \cos\beta\tan(\beta - \alpha) \tag{50}$$

The sign of the derivative of this function with respect to β is the same as the one of :

$$g(\beta) \;=\; \cos(\beta - 2\alpha) \;-\; \cos\beta$$

emphasizing that $\partial\delta/\partial x$ is minimal for $\hat{\beta} = \alpha$ and increases for $\beta > \alpha$ (see Figure 17). This explains why an oscillatory behavior appears on simulations for transients with $\beta < \alpha$ and non zero constant advancement velocity u_0.

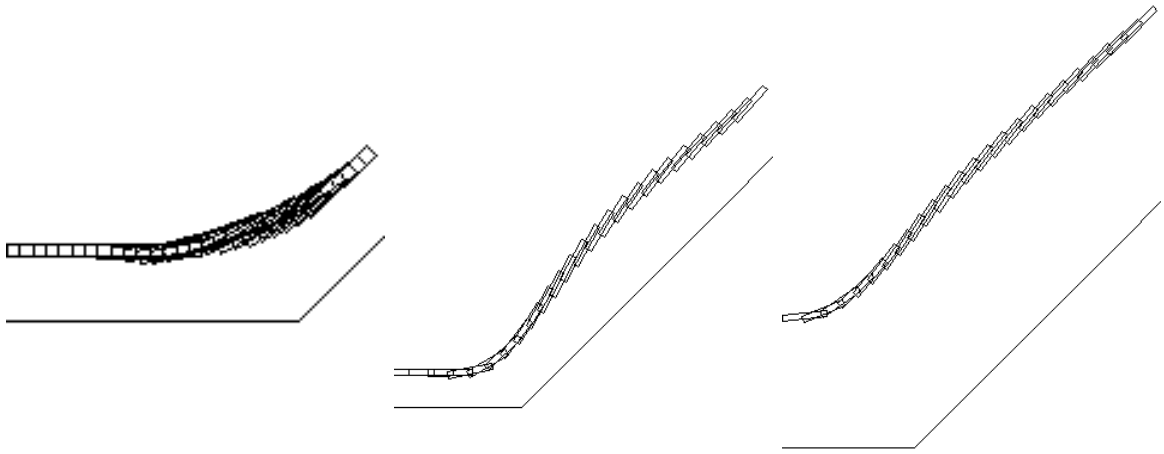
Figures 18 and 19 highlight the effects of the orientation of the rear sensor $(S1)$. A good choice of β_1 is to set it to approximately $\Delta\alpha$ or more. However the corresponding task vector and resulting pitch velocity control are highly chattered as depicted on charts [d] and [f] respectively, even if it doesn't appeared on the resulting vehicle behavior due to its natural damping (charts [a]–[c]). The operational requirements of 400m$\pm$50 is easily respected. If there is any severe constraint on the pitch velocity, this approach would not be convenient.

The second control option is to switch, smoothly whenever possible, from the contour following of the part of the profile that is smooth to a virtual path following the ray of curvature of which can be defined as :

$$R = y_d\left(\frac{\tan\beta_3}{\tan(\Delta\alpha/2)} - 1\right) \quad > 0 \tag{51}$$

leading to a desired pitch velocity $q_d = u_0/R$ to be applied from A to A' on Figure 14. This is equivalent to consider a virtual smooth bottom profile between point B and B' the ray of curvature of which is :

$$R_0 = y_d\frac{\tan\beta_3}{\tan(\Delta\alpha/2)} \quad > 0 \tag{52}$$



(a) u_0 =1m/s, y_d =5m, β_3 =60deg

(b) u_0 =3m/s, y_d =5m, β_3 =40deg

(c) u_0 =3m/s, y_d =20m, β_3 =40deg

Figure 15: Simulation results at various advancement velocities and various reference altitudes when passing an abrupt edge ($r_{min} = 0$) with a break slope value of 45deg. Minimization of a cost-function. For smooth bottom following task, $\beta_1 = \beta_2$ =0deg. For obstacle avoidance, β_3 varies.

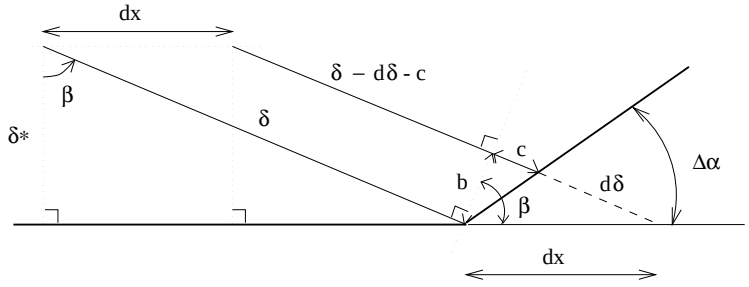


Figure 16: Sensitivity of a measured distance towards a positive slope break

as in (41) and (42), just as in the previous Subsection, but with an opposite sign. The same set of equations (43)–(46) and corresponding comments in Subsection 7.2 are still valid. Once again, detailed simulations confirm the fact that the resulting trajectories are quite smooth, and that they achieve the operational requirements for any load sensor that is considered.

7.4 A profile-varying sea bottom

We would like to achieve a sea bottom following at a desired constant distance (y_d = 10m for Figure 20, y_d = 50m for Figure 21), whatever the initial vehicle situation and the bottom profile are. To achieve this, both contour following

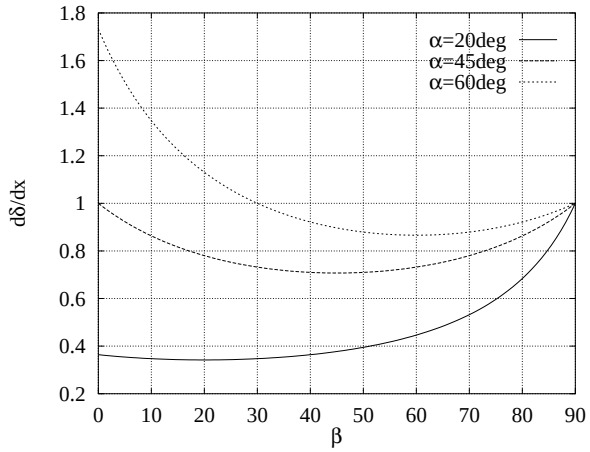


Figure 17: Variations of $\frac{\partial \delta}{\partial x}$ with respect to variables β and α

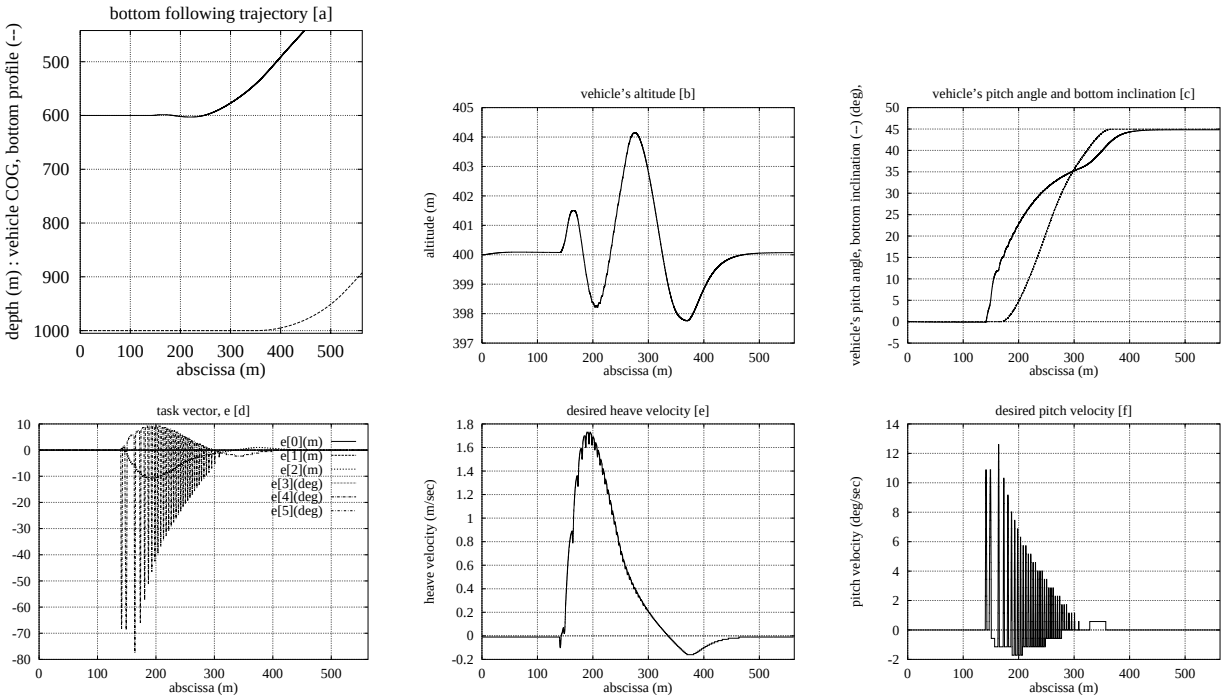


Figure 18: Bottom following of a corner with $y_d > r_{min} > 0$: y_d =400m, u_0 =3m/sec, β_1 =-20deg, β_2 =10deg, β_3 =50deg, $y(0)$ =400m and $\theta(0)$ =0deg, $k_2 = 1$, $k_a = 0.15$ and $k_e = 0.3$

and obstacle avoidance tasks are required. A user-defined bottom profile is defined with slopes that exceed our usual operational constraints in order to show the range of validity of the two approaches proposed here : the break slope values can be positive or negative up to ±100deg, and the slope values

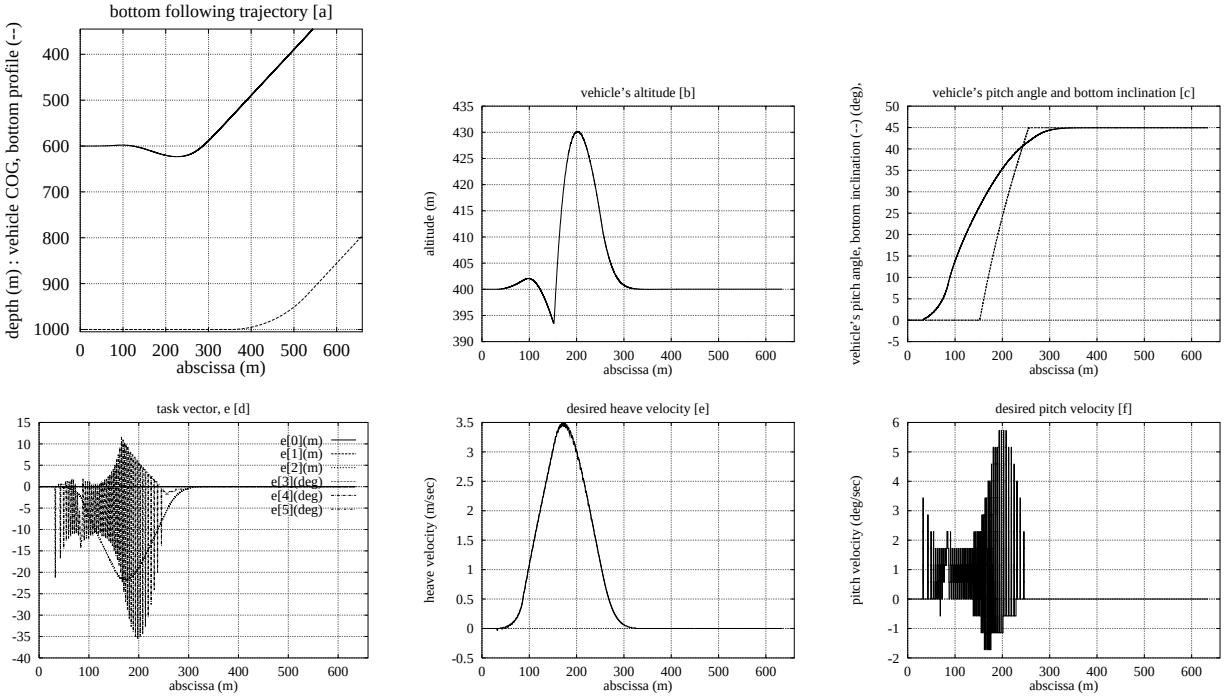


Figure 19: Bottom following of a corner with $y_d > r_{min} > 0$: y_d =400m, u_0 =3m/sec, $\beta_1 = \beta_2$ =0deg, β_3 =50deg, $y(0)$ =400m and $\theta(0)$ =0deg, $k_2 = 1$, $k_a = 0.15$ and $k_e = 0.3$

can be greater than 45deg. It is worth noticing that the vehicle is intended to remain parallel to the bottom profile.

Regarding the approach based on the minimization of a cost-function involving both obstacle avoidance and contour following task-function vectors, Figure 20 shows that the contour following is achieved without collision, and that the approach is robust with respect to noisy sensory data. However, an oscillatory behavior can be noticed after every obstacle avoidance (after abscissa 200m and 320m) involving high break slope values greater than the orientation of the front sensor ($S3$). This can be improved by :

- either increasing the orientation of the sensor (S3) in order to reduce the break on the distance due to the break of the slope
- or adding another sensor with a different orientation affected to the obstacle avoidance task, in order to increase the protection area in front of the vehicle.

However the tuning of the weighting and controller parameters to respect the "tube" constraint of $y_d \pm \Delta y_d$ appears to be difficult, and only test-and-trials can guarantee the desired vehicle's behavior.

Considering now the approach relying on the switching concept, Figure 21 shows that the contour following is achieved as required, and that the approach

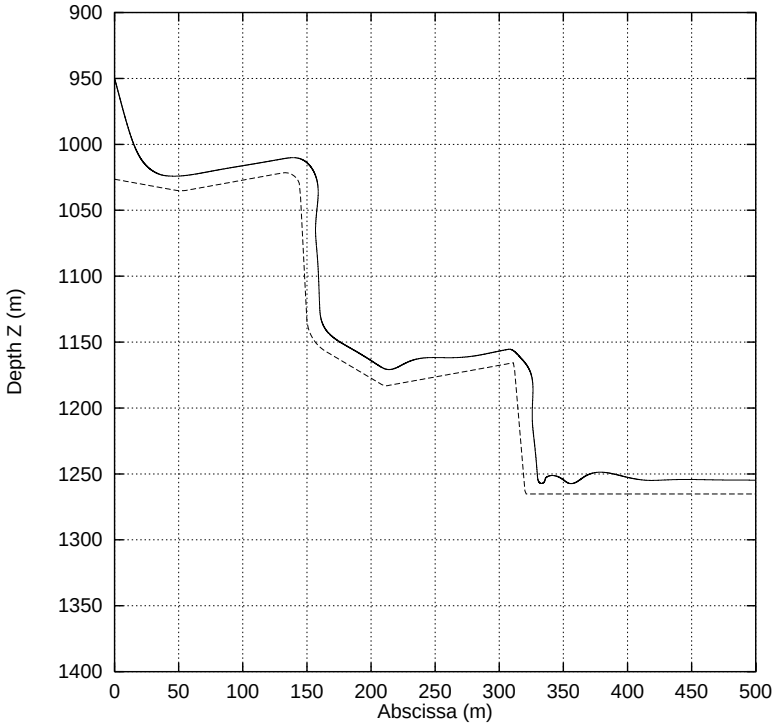


Figure 20: Minimization of a cost-function : following the sea bottom with 3 sensors, 2 applying for contour-following tasks, 1 for obstacle avoidance. Sensors are submitted to Gaussian white noise with a 2-meter standard deviation. $u_o = 3m/s$, y_d = 10m, β_1=-5deg, $beta_2$=5deg, β_3=60deg

is robust with respect to noisy sensory data, thanks to an adequate choice of parameters τ, k_e and k_a. Even corners and edges are well handled at the desired altitude with respect to the profile. Furthermore, an abruptly changing profile do not lead to an oscillatory behavior, again owing to parameters τ, k_e and k_a.

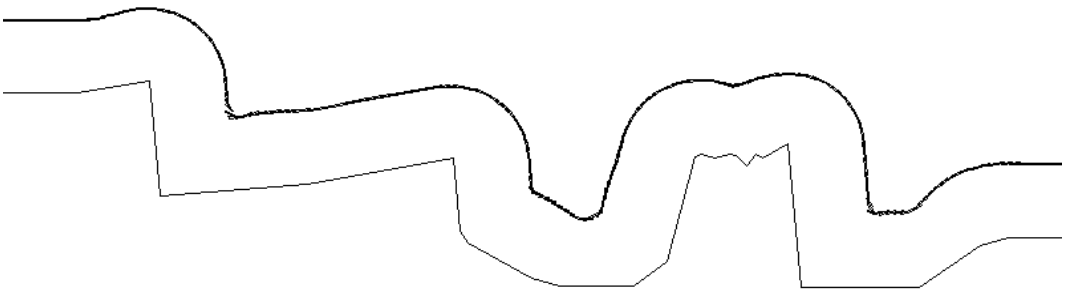


Figure 21: Switching between tasks : following the sea bottom with a 3-sensor configuration, 2 applying for contour-following tasks, 1 for obstacle avoidance. Sensors are submitted to Gaussian white noise with a 2-meter standard deviation. $u_o = 3m/s$, y_d = 50m, β_1=0deg, β_2=0deg, β_3=60deg

8 Conclusion

A global and generic sensor-based control method has been applied to the specific case of sensor-based tasks for fully-actuated AUVs moving in an *a priori* unknown dynamic environment. It is based on the task function approach. In order to face a profile-varying and *a priori* unknown sea bottom including smooth parts, corners, and edges, two concepts of application have been presented and compared.

The first one uses a minimization of a cost-function depending on the obstacle avoidance and contour following task-function vectors. Thanks to an adequate choice of weighting and controller parameters, a complete mission of bottom following can be achieved. However the transients, especially when facing corners, are difficult to quantify and only trial-and-error tests can give the adequate set of tuning parameters. The second one intends to continuously switch between task functions in order to improve and, mainly, control the vehicle's behavior during transient phases when dealing with corners and edges. Both concepts generate a desired velocity screw to be applied on the vehicle. It is worth noticing that a complete study of an adequate altitude sensors configuration is necessary in parallel to the control study.

In order to take into account the vehicle's dynamics, a law level control then can be conveniently chosen among all those that already exist in the literature. Its choice generally depends on the vehicle characteristics and the user's touch and expertise.

The implementation of the concepts developed above is performed presently on the IFREMER's VORTEX vehicle [Simon et al., 1994].

Finally, the approach developed here can be extended to any exteroceptive sensor giving information that only depends on the relative sensor-target situation and the time variable, including cameras.

References

[Ageev, 1994] Ageev, M. (1994). Auv — a precise platform for underwater gravity measurement. In *Proc. of OCEANS'94 OSATES*, pages I–126–129.

[Borrelly and Astraudo, 1992] Borrelly, J.-J. and Astraudo, C. (1992). Simulation of multiprocessor robot controllers. In *Proc. of IEEE Int. Conf. on Robotics and Automation*, pages 573–578.

[Brooks, 1989] Brooks, R. (1989). Robot beings. In *Proc. of Int. Workshop on Intelligent Robots and Systems.*

[Budenske and Gini, 1994] Budenske, J. and Gini, M. (1994). Why is it so difficult for a robot to pass through a doorway using ultrasonic sensors ? In *Proc. of IEEE Int. Conf. on Robotics and Automation*, pages 3124–3129.

[Collar et al., 1994] Collar, P., Babb, R., Michel, J., Brisset, L., and Kilpatrick, I. (1994). Systems research for unmanned autonomous underwater vehicles. In *Proc. of OCEANS'94 OSATES*, pages I–158–163.

[Cristi et al., 1990] Cristi, R., Papoulias, F., and Healey, A. (1990). Adaptive sliding mode control of autonomous underwater vehicles in the dive plane. *IEEE Journal of Oceanic Engineering*, 15(3):152–160.

[de Wit et al., 1993] de Wit, C. C., Williamson, D., and Bachmayer, R. (1993). Performance-oriented robust control for a class of mechanical systems : a study case. In *Proc. of Int. Conf. on Systems, Man and Cybernetics*, pages 51–55.

[Fossen, 1994] Fossen, T. (1994). *Guidance and control of ocean vehicles.* John Wiley and Sons Ltd, Chichester - New-York - Brisbane.

[Fossen and Fjellstad, 1994] Fossen, T. and Fjellstad, O. (1994). Cascaded adaptive control of marine vehicles with significant actuator dynamics. *Journal of Modeling, Identification, and Control*, 15(2):81–91.

[Fryxell et al., 1994] Fryxell, D., Oliveira, P., Pascoal, A., and Silvestre, C. (1994). Integrated design of navigation, guidance and control systems for unmanned underwater vehicles. In *Proc. of OCEANS'94 OSATES*, pages III–105–110.

[Holenstein and Badreddin, 1991] Holenstein, A. and Badreddin, E. (1991). Collision-avoidance in a behavior-based mobile robot. In *Proc. of IEEE Conf. on Robotics and Automation*, pages 898–903.

[Kato, 1994] Kato, N. (1994). *Applications of fuzzy algorithm to guidance and control of underwater vehicles.* edited by J. Yuh, Tsi Press.

[Korba, 1994] Korba, L. (1994). Variable aperture sonar for mobile robots. In *Proc. of IEEE Int. Conf. on Robotics and Automation*, pages 3136–3141.

[Kuc, 1990] Kuc, R. (1990). A spatial sampling criterion for sonar obstacle detection. *IEEE Trans. on PAMI*, 12(7):686–690.

[Kuc and Bozma, 1991] Kuc, R. and Bozma, O. (1991). Building a sonar map in a specular environment using a single mobile sensor. *IEEE Trans. on PAMI*, 13(12):1259–1269.

[Kuc and Siegel, 1987] Kuc, R. and Siegel, M. (1987). Physically based simulation model for acoustic sensor robot navigation. *IEEE Trans. on PAMI*, 9(6):766–778.

[Masson et al., 1994] Masson, D., Sichler, B., Renard, V., Michel, J., and Dias, J. (1994). An autonomous vehicle for geological exploration : sensor payload and mission analysis. In *Proc. of OCEANS'94 OSATES*, pages I–141–146.

[Perrier et al., 1994] Perrier, M., Rigaud, V., de Wit, C. C., and Bachmayer, R. (1994). Performance-oriented robust nonlinear control for subsea robots : experimental validation. In *Proc. of IEEE Int. Conf. on Robotics and Automation*, pages 2095–2099.

[Pissard-Gibollet, 1993] Pissard-Gibollet, R. (1993). *Conception et commande par asservissement visuel d'un robot mobile*. PhD thesis, Ecole Nationale Supérieure des Mines de Paris, Sophia-Antipolis (FRANCE).

[Rigaud, 1990] Rigaud, V. (1990). *Fusion multisensorielle pour le télépilotage assisté par ordinateur d'un véhicule sous-marin*. PhD thesis, Rennes I University, Rennes (FRANCE).

[Samson, 1992] Samson, C. (1992). Path following and time-varying feedback stabilization of a wheeled mobile robot. In *Proc. ICARCV'92*, pages 13.1.1–13.1.5.

[Samson et al., 1990] Samson, C., Espiau, B., and Borgne, M. L. (1990). *Robot control : the task-function approach*. Oxford University Press.

[Santos and Bitmead, 1995] Santos, A. and Bitmead, R. (1995). Nonlinear control for an autonomous underwater vehicle preserving linear design capabilities. In *Proc. of Conference on Decision and Control (Submitted)*.

[Santos and Chaumette, 1992] Santos, A. and Chaumette, F. (1992). Target tracking by visual servoing. *IRISA/INRIA-FRANCE report* **683**, page 50.

[Santos and Simon, 1994] Santos, A. and Simon, D. (1994). Study of altitude sensors configuration. *European MAST II Contract n^0 MAS2-CT92-0028 (Unpublished)*.

[Santos et al., 1994] Santos, A., Simon, D., and Rigaud, V. (1994). A sensor-based high-level control approach for autonomous underwater vehicles (auvs). In *Proc. of Intelligent Robotics Systems*, pages 170–177.

[Santos et al., 1995] Santos, A., Simon, D., and Rigaud, V. (1995). Sensor-based control of a class of under-actuated autonomous underwater vehicles. In *Proc. of IFAC Control Applications in Marine Systems*, volume to appear.

[Shabana, 1989] Shabana, A. (1989). *Dynamics of multibody systems.* John Wiley and Sons, Ltd.

[Simon et al., 1994] Simon, D., Coste-Maniere, E., Pissard, R., Rigaud, V., Perrier, M., and Peuch, A. (1994). A reactive approach to underwater-vehicle : the mixed orccad/pirat programming of the vortex vehicle. In *Proc. of International Advanced Robotics Programs.*

[Slotine and Li, 1991] Slotine, J.-J. E. and Li, W. (1991). *Applied nonlinear control.* Prentice-Hall International Editions.

[Soldo, 1990] Soldo, M. (1990). Reactive and preplanned control in a mobile robot. In *Proc. of IEEE Int. Conf. on Robotics and Automation*, pages 1128–1132.

[Zapata and Lépinay, 1991] Zapata, R. and Lépinay, P. (1991). Fast mobile robots in unstructured environments. *Lecture notes in Computer Science : Geometric Reasoning for perception and action*, 708:94–105.

Unité de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unité de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unité de recherche INRIA Rhône-Alpes, 46 avenue Félix Viallet, 38031 GRENOBLE Cedex 1
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
ISSN 0249-6399