# Robust techniques of automatic control for mobile robotic systems

Juan Luis Rosendo

# THÈSE DE DOCTORAT

Préparée en co-tutelle entre
L 'ECOLE NATIONALE SUPERIEURE DE TECHNIQUES AVANCEES BRETAGNE
COMUE UNIVERSITE BRETAGNE LOIRE
ECOLE DOCTORALE N ° 601
Mathématiques et Sciences et Technologies de l'Information et de la Communication
Et
L'UNIVERSIDAD NACIONAL DE LA PLATA (UNLP)

Spécialité : Robotique
Par

« **Juan Luis ROSENDO** »

## « Techniques robustes pour le contrôle automatique des systèmes robotiques »

**Thèse présentée et soutenue à** LA PLATA , le 22 mars 2019
**Unité de recherche : Lab-STICC UMR CNRS6285**

**Rapporteurs avant soutenance :**
M. Gerardo G.ACOSTA, Professeur, UNCPBA - CONICET, Buenos Aires
M. Carlos SORIA, directeur de Recherche INAUT - CONICET, San Juan

**Composition du jury :**

| | |
|---|---|
| Président : | M. Gerardo G.ACOSTA, Professeur, UNCPBA - CONICET, Buenos Aires |
| Examinateurs : | M. Ignacio A. MAS, Assistant Professor, Instituto Tecnologico de Buenos Aires - CONICET |
| | M. Benoît CLEMENT, Professeur, ENSTA Bretagne Brest, directeur de thèse |
| | M. Fabricio GARELLI, Professeur, Universidad Nacional de la Plata, co-directeur de thèse |
| | M. Vincent CREUZE, Maître de conférences, HDR, Université de Montpellier |

Invité(s)

M. Hernán DE BATTISTA, Associate Professor, Universidad Nacional de la Plata, co-encadrant

M. Carlos SORIA, directeur de Recherche INAUT - CONICET, San Juan

# ACKNOWLEDGEMENT

The first and greatest gratitude is for my family who has always been by my side pushing me and enduring the sacrifices that have involved this work.

I can do no other than thanks my directors Fabricio, Benoit and Hernán, who have been able to advise me on the way I have traveled.

In particular, I have to thank all those people with whom I had the opportunity to work both at ENSTA and UNLP, with whom it has been a pleasure to share time and anecdotes.

Finally, I must thank the agencies that have financed this research and without which this research could not have been carried out CONICET, Eiffel Scholarship, UNLP, and ENSTA Bretagne.

# TABLE OF CONTENTS

CHAPITRE 1

# INTRODUCTION

This chapter aims to give the motivations for this work and a general overview of the state of the current robotics developments. Its usual classifications are presented, allowing to group robots according to their common characteristics and particular applications, what encompass developed or developing technologies.

## 1.1 Brief introduction to robotics

The term robot refers to systems with a certain degree of autonomy which perform tasks, in a greater or lesser degree predefined by an user. Most of these systems, or at least with the definition taken here, have a physical structure to interact with the environment around them. This interaction with the environment together with their degree of autonomy, require a control system capable of processing the stimulus of the environment, and respond based on them, according to a planning stage. Again, at the time of interaction, the control system will exert control over the system actuators (motors, pistons, etc) and will receive information of the environment through the most diverse variety of sensors available. Therefore, it can be recognized that robotics is an interdisciplinary subject concerning the areas of mechanics, control, computers, and electronics.
Considering robotics as a multidisciplinary subject, the robotic systems can be classified in different ways. Next, two possible classifications will be presented. These ones are closest to the objectives of the work and for author's discretion are broad enough to include the diversity of existing robots.

### 1.1.1 Fixed vs. mobile robotics

The **first classification** of these systems is approached from the mechanical point of view and the implications that it has on the control technologies and final operations.

According to this criterion, robots can be divided in those with fixed base (fixed robotics) and those with mobile base (mobile robotics), see Fig. 1.1. In the following, the geometrical features of the two classes are presented.

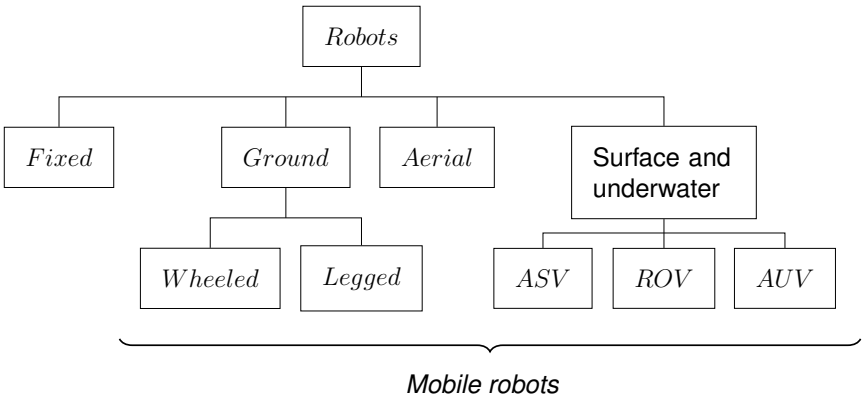



FIGURE 1.1 – Mechanical classification of robots

— **Fixed robotics :** this classification is composed of the robots called manipulators. The mechanical structure of a manipulator robot consists of a sequence of rigid bodies (links) interconnected by means of articulations (joints) ; a manipulator is characterized by an arm that ensures mobility, a wrist that confers dexterity, and an end-effector that performs the task required to the robot, see Fig. 1.2. The fundamental structure of a manipulator is the serial or open kinematic chain.

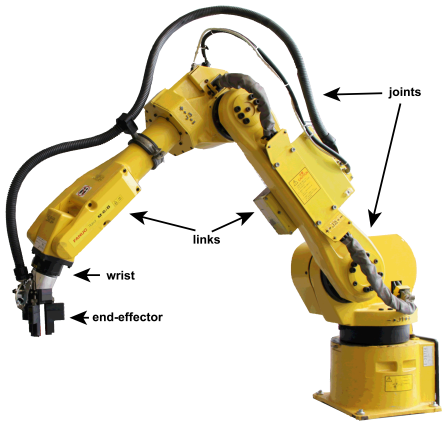


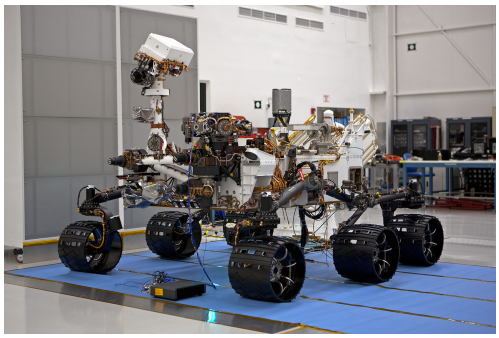

FIGURE 1.2 – Manipulator

— **Mobile robotics :** The main feature of mobile robots is the presence of a mobile base which allows the robot to move freely in the environment. Unlike manipulators, such robots are mostly used in service applications, where extensive, autonomous motion capabilities are required. From a mechanical viewpoint, a mobile robot consists of one or more rigid bodies equipped with a locomotion system. This kind of robots are usually classified according to the environment for which they are designed :

— **Ground robotics :** This type of robots have the greatest diversity in terms of their form of locomotion, however two large groups can be distinguished :

— **Wheeled mobile robots :** typically consist of a rigid body (base or chassis) and a system of wheels which provide motion with respect to the ground. Other rigid bodies (trailers), also equipped with wheels, may be connected to the base by means of revolute joints, see Fig. 1.3.

(a) Pionner 3-AT



(b) Mars Science Laboratory (Curiosity)

FIGURE 1.3 – Examples of ground robots

— **Legged mobile robots :** made of multiple rigid bodies, interconnected by prismatic joints or, more often, by revolute joints. Some of these bodies form lower limbs, whose extremities (feet) periodically come in contact with the ground to realize locomotion. There is a large variety of mechanical structures in this class, whose design is often inspired by the study of living organisms (biomimetic robotics) : they range from biped humanoids to hexapod robots aimed at replicating the biomechanical efficiency of insects. In Fig. 1.4, several commercial robots of this kind can be observed.

FIGURE 1.4 – Mobile legged robots

— **Aerial robotics :** In this case it is possible to find the robotic bases with flight capacity, self-propelled or glide. In aerospace jargon, robotic flying machines are commonly referred to as unmanned aerial vehicles (UAVs), while the entire infrastructures, systems and human components required to operate such machines for a given goal are often called unmanned aerials systems (UASs).


FIGURE 1.5 – UAV Vigia 2B of the Argentine Air Force

— **Surface and underwater robotics :** In this subdivision it is possible to find mobile bases with the capacity to move over (ASV, autonomous surface robots) and under the water surface. Referring to the underwater vehicles that will occupy a large part of this work, we find again a division according to its operation :
  — **remotely operated vehicles (ROVs)**, physically connected via a tether to receive power and data. These vehicles are commanded by a human and usually are used only for the shallowest installations. Subsea systems require extensive work capability during installation, and need frequent inspection and intervention to support drilling operations, actuate valves, repair or replace subsea components, and to accomplish a variety of tasks required to maintain production rates and product quality. The effective-

ness of using ROVs decreases with depth mainly due to the cost increase and the difficulties of handling the long tether. In Fig. 1.6, is possible to see examples of these robots.

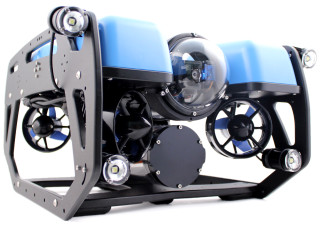
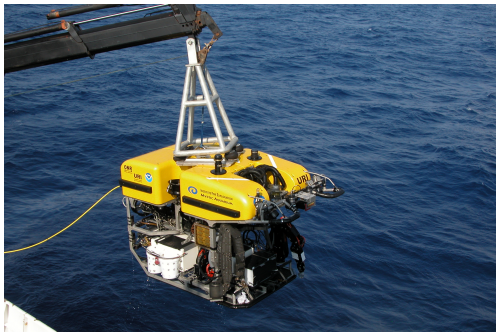


Image taken from :
https ://www.bluerobotics.com

(a) Bluerov2.



(b) ROV Hercules

FIGURE 1.6 – Examples of ROV robots

— **Autonomous underwater vehicles (AUVs)** are freeswimming unoccupied underwater vehicles that can overcome the limitations imposed by ROV tethers for some tasks. Such vehicles carry their own energy supplies and communicate only through acoustics and optical links. Fig.1.7 shows an example of this kind of robots : the SPARUS II AUV developed by iquarobotics. Limited communications require these vehicles to operate independently of continuous human control, in many cases the vehicles operate completely autonomously. AUVs are currently used for scientific survey tasks, oceanographic sampling, underwater archeology and under-ice survey. Military applications, such as mine detection and landing site survey, are presently operational, and more ambitious applications such as long-term undersea surveillance are in engineering development. Presently, AUVs are incapable of sampling or manipulations tasks like those done routinely by ROVs, as typical work environments tend to be complex and challenging even to skilled human pilots.

### 1.1.2 Industrial & advanced robotics

A **second classification** of robots considers the potential use of robots in different fields of application and the degree of development of the technologies that are used in these applications :

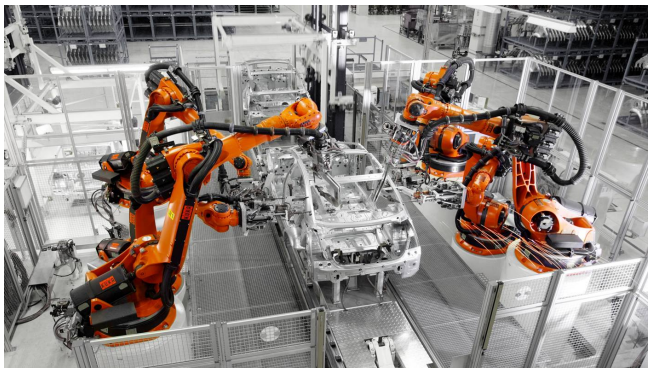

Image taken from : http ://iquarobotics.com/sparus-ii-auv

FIGURE 1.7 – AUV Sparus II

— **industrial robotics :** this classification concerns about robot design, control and applications in industry, which have by now reached the level of a mature technology, see Fig. 1.8. The connotation of a robot for industrial applications is that of operating in a structured environment whose geometrical or physical characteristics are mostly known a priori. Hence, limited autonomy is required. Typical applications include :
  — palletizing (placing objects on a pallet in an ordered way),
  — warehouse loading and unloading,
  — mill and machine tool tending,
  — part sorting,
  — packaging.
  — object inspection,
  — painting and welding,

  In these applications, besides robots, Automated Guided Vehicles (AGV) are utilized which ensure handling of parts and tools around the shop floor from one manufacturing cell to the next. Compared with the traditional fixed guide paths for vehicles (inductive guide wire, magnetic tape, or optical visible line), modern AGVs utilize high-tech systems with onboard microprocessors and sensors (laser, odometry, GPS). It allows their localization within the plant layout and manage their work flow, making possible their complete integration

in flexible manufacturing systems. The mobile robots employed in advanced applications can be considered as the natural evolution of the AGV systems, as far as enhanced autonomy is concerned.



(a) Robotic assembly line.
Image taken from : https ://www.kuka.com

(b) AGV system.
Image taken from : https ://www.conductix.us

FIGURE 1.8 – Examples of industrial robotic

— **advanced robotics :** this expression usually refers to the science studying robots with marked characteristics of autonomy, operating in scarcely structured or unstructured environments, whose geometrical or physical characteristics would not be known a priori. The advanced robotics is in full development. However, today it is possible to find robots that take care of domestic tasks, that collaborate with care tasks, that explore inaccessible places, etc. These technologies are not yet fully closed and many of these products are in stages of testing or prototyping. There are many motivations which strongly encourage advances in

knowledge within this field. They range from the need for automata whenever human operators are not available or are not safe (field robots), to the opportunity of developing products for potentially wide markets which are aimed at improving quality of life (service robots).

## 1.2 About this work

From what has been previously shown, it can be seen that the largest areas of growth in applications and the consequent new problems are in the area of mobile robotics and its potential application to the areas of service and field robotics.

While the development of mobile robotics has a long history since the 60s [1], even problems such as track monitoring, obstacle avoidance, energy consumption and incomplete modeling of systems continue affecting their applications. These problems are of interest since their solution, even in partial mode, can lead to higher quality standards. Furthermore, solutions with robust features will expand even more the range of possible applications in the future.

The motivation of this Thesis lies in working on these problems, from the point of view of control theory, with the aim of contributing to its partial resolution. Also in a broader aspect this Thesis finds motivation in the vacancy of the subject at the UNLP, and its possible implications for the development of this line of research in the future.

In this work the problems addressed are focused on the special case of autonomous systems with constraints. It is known that restrictions present in this systems affect their desired behaviors. In this Thesis first through the implementation of auxiliary control loops and then developing tuning methods for main controllers, these constraint effects will be partially mitigated.

The material presented along the chapters of this work is organized in terms of the degree of freedom in the control tuning and design. In each chapter a first part introducing the necessary theory to approach each proposal will be carried out, and then a case study will be exposed with its corresponding results and conclusions.

After introducing some fundamental concepts and main problems that affect the mobile robotic in chapter 2, chapters 3 and 4 will address the problem of mitigating the constraints effects through auxiliary loops.

In chapter 3 the problem of path following under actuator saturation is interpreted as a problem of inputs constraints. To mitigate their effects, a sliding mode based tech-

nique is proposed to modify the speed of movement along the road, making the robot advance compatible with the restrictions on actuators. As a case study, this technique is implemented on real AUV achieving an overall path time improvement in presence of actuator constraints. For the validation of this technique and others throughout this Thesis, a detailed modeling on the Ciscrea AUV is performed and validated experimentally. The use of this vehicle has been thanks to the collaboration established between research groups at the ENSTA Bretagne and UNLP in the framework of an Eiffel scholarship.

In chapter 4, the approach of using an auxiliary loop to complement the function of a base controller is continued. In this case the problem of obstacle avoidance in mobile robots over restricted paths is addressed. It is interpreted as a problem of constraints in the output of these systems. Through the development of the Collision Avoidance Speed Adaption (CASA) method, a solution to this problem by modifying the robot's progress profile in collision situations is carried out. In this chapter, two study cases have been taken : the case of a mobile land robot and the one of the Ciscrea AUV.

As in chapters 3 and 4, the proposals presented depended on the base controllers to guarantee the desired behavior, in chapter 5 a robust tuning technique for main controllers under PID structural constraints is proposed. The robust approach for tuning is justified in the application to mobile robots, since these usually suffer from modeling uncertainty and they are prone to disturbances due to their working environments. As a case study, the yaw control direction of the Ciscrea AUV is taken.

Chapter 6 seeks to obtain a methodology based on global optimization and interval techniques for the tuning of robust controllers for non-linear systems such as mobile robots. Unlike chapter 5, here it is assumed that the main control can be changed and the proposed methodology is based on the tuning of sliding mode controls. Through the use of interval techniques, maps called subpavings are generated providing information about the regions of the state-space where these controllers work in a guaranteed way. The methodology developed in this chapter can be seen as a form of tuning for the auxiliary loops presented in chapters 3 and 4.

Finally, chapter 7 presents a brief synthesis of the advances made in this Thesis as well as possible extensions of the works presented here.

CHAPITRE 2

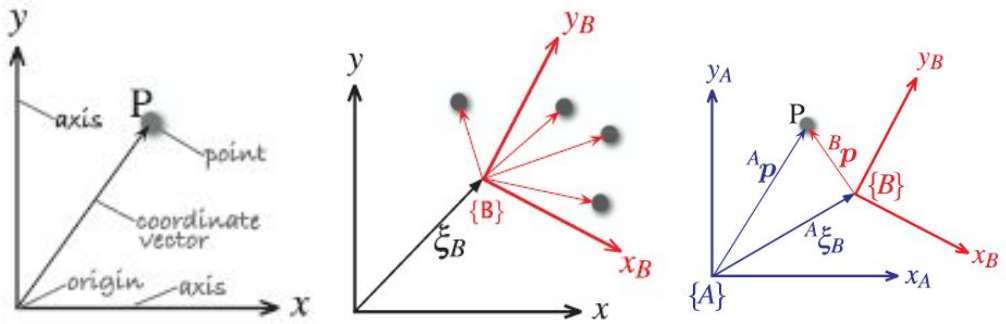
# FUNDAMENTAL ROBOTIC NOTIONS

There are a number of topics necessary for the development of any study on robotics. In this chapter the necessary ones for the development of the proposed control strategies will be approached. In addition, a brief introduction to the problems that will be addressed throughout the work is carried out, placing emphasis on application cases that would benefit from the techniques developed.

## 2.1 Position and orientation

A fundamental requirement in robotics is to represent the position and orientation of objects in an environment [2]. Such objects include robots, cameras, work-pieces, obstacles and paths. A point in space can be described by a coordinate vector, also known as a bound vector, as shown in Fig. 2.1a. The vector represents the displacement of the point with respect to some reference coordinate frame. A coordinate frame is a set of orthogonal axes which intersect at a point known as the origin. More frequently it is necessary to consider a set of points that comprise some object. It is assumed that the object is rigid and that its constituent points maintain a constant relative position with respect to the object's coordinate frame as shown in Fig. 2.1b. Instead of describing the individual points we describe the position and orientation of the object by the position and orientation of its coordinate frame. A coordinate frame is labelled, $\{B\}$ in this case, and its axis labels $x_B$ and $y_B$ adopt the frame's label as their subscript.

The position and orientation of a coordinate frame is known as its pose and is shown graphically as a set of coordinate axes. The relative pose of a frame with respect to a reference coordinate frame is denoted by the symbol $\xi$. Figure 2.1c shows two frames $\{A\}$ and $\{B\}$ and the relative pose ${}^A\xi_B$ which describes $\{B\}$ with respect to $\{A\}$. The leading superscript denotes the reference coordinate frame and the subscript denotes the frame being described. We could also think of ${}^A\xi_B$ as describing some motion,




(a) (b) (c)

FIGURE 2.1 – Coordinate vectors

imagine taking $\{A\}$ and applying a displacement and a rotation so that it is transformed to $\{B\}$. If the initial superscript is missing we assume that the change in pose is relative to the world coordinate frame denoted $O$. The point $P$ in Fig. 2.1c can be described with respect to either coordinate frame. Formally we express this as :

$$ {}^A\mathbf{p} = {}^A\,\xi_B \cdot {}^B\,\mathbf{p} \tag{2.1} $$

where the right-hand side expresses the motion from $\{A\}$ to $\{B\}$ and then to $P$. The operator $\cdot$ transforms the vector, resulting in a new vector that describes the same point but with respect to a different coordinate frame. An important characteristic of relative poses is that they can be composed or compounded. Consider the case shown in Fig. 2.2. If one frame can be described in terms of another by a relative pose then they can be applied sequentially

$$ {}^A\xi_C = {}^A\,\xi_B \oplus {}^B\,\xi_C \tag{2.2} $$

which says, in words, that the pose of $\{C\}$ relative to $\{A\}$ can be obtained by compounding the relative poses from $\{A\}$ to $\{B\}$ and $\{B\}$ to $\{C\}$. We use the operator $\oplus$ to indicate composition of relative poses. For this case the point $P$ can be described :

$$ {}^A\mathbf{p} = ({}^A\xi_B \oplus {}^B\,\xi_C) \cdot {}^C\,\mathbf{p} \tag{2.3} $$

Later in this section we will convert these abstract notions of $\xi$, $\cdot$ and $\oplus$ into standard mathematical objects and operators.

In these examples are shown 2-dimensional coordinate frames. This is appropriate

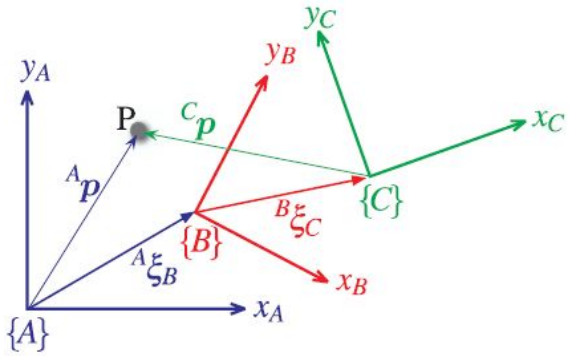



FIGURE 2.2 – Relative poses

for a large class of robotics problems, particularly for mobile robots which operate in a planar world. For other problems it is required 3-dimensional coordinate frames to describe objects in our 3-dimensional world such as the pose of a flying or underwater robot or the end of a tool carried by a robot arm.

### 2.1.1 Pose in 2-Dimensions

We use a coordinate frame with orthogonal axes denoted $x$ and $y$ and typically drawn with the x-axis horizontal and the y-axis vertical. The point of intersection is called the origin. Unit-vectors parallel to the axes are denoted $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$. A point is represented by its x- and y-coordinates $(x, y)$ or as a bound vector

$$\mathbf{p} = x\hat{\mathbf{x}} + y\hat{\mathbf{y}} \tag{2.4}$$

Fig. 2.3 shows a coordinate frame $\{B\}$ that we wish to describe with respect to the reference frame $\{A\}$. We can see clearly that the origin of $\{B\}$ has been displaced by the vector $\mathbf{t} = (x, y)$ and then rotated counter-clockwise by an angle $\theta$. A concrete representation of pose is therefore the 3-vector ${}^A\xi_B \equiv (x, y, \theta)$. Unfortunately this representation is not convenient for the computational calculation, instead we will use a different way of representing rotation.

The approach is to consider an arbitrary point $\mathbf{P}$ with respect to each of the coordinate frames and to determine the relationship between ${}^A\mathbf{p}$ and ${}^B\mathbf{p}$. Referring again to Fig. 2.3 we will consider the problem in two parts : rotation and then translation.

To consider just rotation we create a new frame $\{V\}$ whose axes are parallel to those of $\{A\}$ but whose origin is the same as $\{B\}$, see Fig. 2.4.It is possible to express

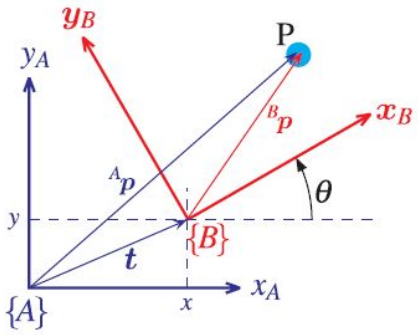



FIGURE 2.3 – Rotated and translated coordinate frame

the point $\mathbf{P}$ with respect to $\{V\}$ in terms of the unit-vectors that define the axes of the frame :

$$ {}^V p = {}^V x \hat{\mathbf{x}}_V + {}^V y \hat{\mathbf{y}}_V = \begin{pmatrix} \hat{\mathbf{x}}_V & \hat{\mathbf{y}}_V \end{pmatrix} \begin{pmatrix} {}^V x \\ {}^V y \end{pmatrix} \tag{2.5} $$

The coordinate frame $\{B\}$ is completely described by its two orthogonal axes which we represent by two unit vectors

$$ \hat{\mathbf{x}}_B = \cos\theta \hat{\mathbf{x}}_V + \sin\theta \hat{\mathbf{y}}_V \tag{2.6} $$

$$ \hat{\mathbf{y}}_B = -\sin\theta \hat{\mathbf{x}}_V + \cos\theta \hat{\mathbf{y}}_V \tag{2.7} $$

which can be factorized into matrix form as :

$$ \begin{pmatrix} \hat{\mathbf{x}}_B & \hat{\mathbf{y}}_B \end{pmatrix} = \begin{pmatrix} \hat{\mathbf{x}}_V & \hat{\mathbf{y}}_V \end{pmatrix} \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \tag{2.8} $$

It is possible to represent the point $P$ with respect to $\{B\}$ as :

$$ {}^B p = {}^B x \hat{\mathbf{x}}_B + {}^B y \hat{\mathbf{y}}_B = \begin{pmatrix} \hat{\mathbf{x}}_B & \hat{\mathbf{y}}_B \end{pmatrix} \begin{pmatrix} {}^B x \\ {}^B y \end{pmatrix} \tag{2.9} $$

substituting Eq.2.8, it is possible to write :

$$ {}^B p = \begin{pmatrix} \hat{\mathbf{x}}_V & \hat{\mathbf{y}}_V \end{pmatrix} \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} {}^B x \\ {}^B y \end{pmatrix} \tag{2.10} $$

Now by equating the coefficients of the right-hand sides of Eq.2.5 and Eq. 2.10 is

possible to get :

$$\begin{pmatrix} {}^{V}x \\ {}^{V}y \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} {}^{B}x \\ {}^{B}y \end{pmatrix} \tag{2.11}$$

which describes how points are trasnformed from frame $\{B\}$ to frame $\{V\}$ when the frame is rotated. This type of matrix is known as a rotation matrix denoted ${}^{V}\mathbf{R}_{B}$.

$$\begin{pmatrix} {}^{V}x \\ {}^{V}y \end{pmatrix} = {}^{V}\mathbf{R}_{B} \begin{pmatrix} {}^{B}x \\ {}^{B}y \end{pmatrix} \tag{2.12}$$

The rotation matrices have some special properties :

— they are orthonormal.

— its determinant is $+1$.

— its inverse is the same as the trasnpose $\mathbf{R}^{-1} = \mathbf{R}^{T}$.

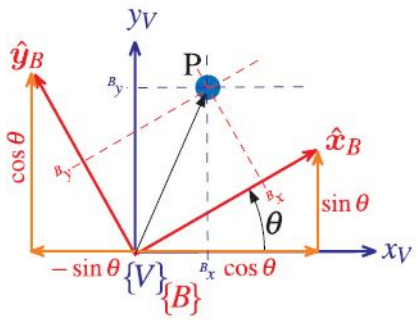



FIGURE 2.4 – Auxiliary coordinate frame

The second part of representing pose is to account for the translation between the origins of the frames shown in Fig.2.3. Since the axes $\{V\}$ and $\{A\}$ are parallel this is simply vectorial addition

$$\begin{pmatrix} {}^{A}x \\ {}^{A}y \end{pmatrix} = \begin{pmatrix} {}^{V}x \\ {}^{V}y \end{pmatrix} + \begin{pmatrix} x \\ y \end{pmatrix} \tag{2.13}$$

$$= \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} {}^{B}x \\ {}^{B}y \end{pmatrix} + \begin{pmatrix} x \\ y \end{pmatrix} \tag{2.14}$$

$$= \begin{pmatrix} \cos\theta & -\sin\theta & x \\ \sin\theta & \cos\theta & y \end{pmatrix} \begin{pmatrix} {}^{B}x \\ {}^{B}y \\ 1 \end{pmatrix} \tag{2.15}$$

or in a more compact form

$$\begin{pmatrix} {}^{A}x \\ {}^{A}y \\ 1 \end{pmatrix} = \begin{pmatrix} {}^{A}\mathbf{R}_B & \mathbf{t} \\ \mathbf{0}_{1x2} & 1 \end{pmatrix} + \begin{pmatrix} {}^{B}x \\ {}^{B}y \\ 1 \end{pmatrix} \tag{2.16}$$

where $\mathbf{t} = (x, y)$ is the translation of the frame and the orientation is ${}^{A}\mathbf{R}_B$. The coordinate vectors for point $P$ are now expressed in homogenous form and it is written :

$${}^{A}\tilde{\mathbf{p}} = \begin{pmatrix} {}^{A}\mathbf{R}_B & \mathbf{t} \\ \mathbf{0}_{1x2} & 1 \end{pmatrix} {}^{B}\tilde{\mathbf{p}} = {}^{A}\mathbf{T}_B {}^{B}\tilde{\mathbf{p}} \tag{2.17}$$

and ${}^{A}\mathbf{T}_B$ is referred to as a homogeneous transformation. By comparison with Eq.2.1 it is clear that ${}^{A}\mathbf{T}_B$ represents relative pose

$$\xi(x, y, \theta) \equiv \begin{pmatrix} \cos\theta & -\sin\theta & x \\ \sin\theta & \cos\theta & y \\ 0 & 0 & 1 \end{pmatrix} \tag{2.18}$$

**Note 1.** *A vector* $\mathbf{p} = (x, y)$ *is written in homogeneous form as* $\tilde{\mathbf{p}} \in \mathbb{P}^2$,$\tilde{\mathbf{p}} = (x_1, x_2, x_3)$ *where* $x = x_1/x_3$, $y = x_2/x_3$ *and* $x_3 \neq 0$.

### 2.1.2 Pose in 3-Dimensions

The 3-dimensional case is an extension of the 2-dimensional case discussed previously. It adds an extra coordinate axis, typically denoted by z, that is orthogonal to both the x- and y-axes. The direction of the z-axis obeys the right-hand rule and forms a right-handed coordinate frame. Unit vectors parallel to the axes are denoted $\hat{\mathbf{x}}$, $\hat{\mathbf{y}}$ and $\hat{\mathbf{z}}$ such that

$$\hat{\mathbf{z}} = \hat{\mathbf{x}} \times \hat{\mathbf{y}}, \quad \hat{\mathbf{x}} = \hat{\mathbf{y}} \times \hat{\mathbf{z}}, \quad \hat{\mathbf{y}} = \hat{\mathbf{z}} \times \hat{\mathbf{x}}, \tag{2.19}$$

A point $P$ is represented by its $x$, $y$ and $z$ coordinates $(x, y, z)$ or as a bound vector

$$p = x\hat{\mathbf{x}} + y\hat{\mathbf{y}} + z\hat{\mathbf{z}} \tag{2.20}$$

Fig. 2.5 shows a coordinate frame $\{B\}$, suppose that it is desired to describe with respect to the reference frame $\{A\}$. It is possible to see that the origin of $\{B\}$ has been displaced by the vector $\mathbf{t} = (x, y, z)$ and then rotated. Just as for the 2-dimensional

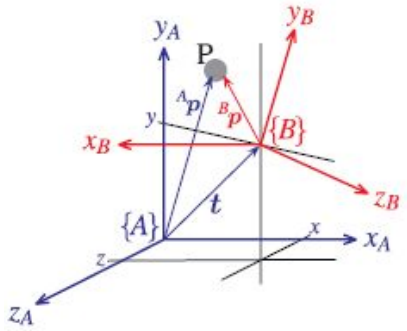



FIGURE 2.5 – 3D coordinate frames

case,the chosen approach is to consider an arbitrary point P with respect to each of the coordinate frames and to determine the relationship between ${}^A\mathbf{p}$ and ${}^B\mathbf{p}$. It is considered again the problem in two parts : rotation and then translation.

### Representing Orientation in 3-Dimensions

To describe the orientation of the pair of coordinate frames of Fig. 2.5 one with respect to the other, it is possible to imagine picking up frame $\{A\}$ and rotating it until it looked just like frame $\{B\}$. Euler's rotation theorem states that any rotation can be considered as a sequence of rotations about different coordinate axes.

**Theorem 2.1.1.** *Euler's rotation theorem : Any two independent orthonormal coordinate frames can be related by a sequence of rotations (not more than three) about coordinate axes, where no two successive rotations may be about the same axis.*

The issue of rotation is that the sequence of two rotations applied in different orders do not lead to the same orientation, thus the order in which are applied is determinant. The implication for the pose algebra is that the $\oplus$ operator is not commutative.

### Orthonormal Rotation Matrix

Just as for the 2-dimensional case it is possible to represent the orientation of a coordinate frame by its unit vectors expressed in terms of the reference coordinate frame. Each unit vector has three elements and they form the columns of a $3 \times 3$

orthonormal matrix ${}^{A}\mathbf{R}_{B}$.

$$\begin{pmatrix} {}^{A}x \\ {}^{A}y \\ {}^{A}z \end{pmatrix} = {}^{A}\mathbf{R}_{B} \begin{pmatrix} {}^{B}x \\ {}^{B}y \\ {}^{B}z \end{pmatrix} \tag{2.21}$$

which rotates a vector defined with respect to frame $\{B\}$ to a vector with respect to $\{A\}$. The orthonormal rotation matrices for rotation of $\theta$ about the $x$,$y$ and $z$ axes are

$$R_x(\theta) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{pmatrix} \tag{2.22}$$

$$R_y(\theta) = \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix} \tag{2.23}$$

$$R_z(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{2.24}$$

**Three-Angle representations**

Euler's rotation theorem requires successive rotation about three axes such that no two successive rotations are about the same axis. There are two classes of rotation sequence : Eulerian and Cardanian. The Eulerian type involves repetition, but not successive, of rotations about one particular axis : XYX, XZX, YXY, YZY, ZXZ, or ZYZ. The Cardanian type is characterized by rotations about all three axes : XYZ, XZY, YZX, YXZ, ZXY, or ZYX. In common usage all these sequences are called Euler angles and there are a total of twelve to choose from.

Two common sequences used are :

— The ZYZ sequence, is commonly used in aeronautics and mechanical dynamics. This sequence presents the particularity that its inverse has a singularity at $\theta = 0$.

$$\mathbf{R} = \mathbf{R}_z(\phi)\mathbf{R}_y(\theta)\mathbf{R}_z(\psi) \tag{2.25}$$

— The XYZ sequence, roll-pitch-yaw angle sequence, is more intuitive when des-

cribing the attitude of vehicles such as ships, aircrafts and cars. Roll, pitch and yaw (also called bank, attitude and heading) refer to rotations about the $x$, $y$, $z$ axes, respectively. This XYZ angle sequence, technically Cardan angles, are also known as Tait-Bryan angles or nautical angles. For aerospace and ground vehicles the x-axis is commonly defined in the forward direction, z-axis downward and the y-axis to the right-hand side.

$$\mathbf{R} = \mathbf{R}_x(\theta_r)\mathbf{R}_y(\theta_p)\mathbf{R}_z(\theta_y) \tag{2.26}$$

For this sequence when its inverse is calculated the singularity is present at $\theta_p = \pm\frac{\pi}{2}$.

In any case, the 3-vector $\Gamma = (\phi, \theta, \psi)$ is know as the Euler angles.

A fundamental problem with the three-angle representations just described is singularity. This occurs when the rotational axis of the middle term in the sequence becomes parallel to the rotation axis of the first or third term. Singularities are an unfortunate consequence of using a minimal representation. To eliminate this problem it is needed to adopt different representations of orientation.

### Unit Quaternions

The quaternion is an extension of the complex number – a hyper-complex number – and is written as a scalar plus a vector

$$\mathring{q} = s + \mathbf{v} = s + v_1\mathbf{i} + v_2\mathbf{j} + v_3\mathbf{k} \tag{2.27}$$

where $s \in \mathbb{R}$, $\mathbf{v} \in \mathbb{R}^3$ and the orthogonal complex numbers $i$, $j$ and $k$ are defined such that

$$\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1 \tag{2.28}$$

The quaternion is denoted as

$$\mathring{q} = s < v_1, v_2, v_3 > \tag{2.29}$$

Quaternions are computationally straightforward and widely used for robotics, computer vision, computer graphics and aerospace inertial navigation applications. To represent rotations is possible to use unit-quaternions. These are quaternions of unit

magnitude, that is, those for which $|\mathring{q}| = 1$ or $s^2 + v_1^2 + v_2^2 + v_3^2 = 1$. The unit-quaternion has the special property that it can be considered as a rotation of $\theta$ about the unit vector $\hat{\mathbf{n}}$ which are related to the quaternion components by

$$s = \cos\frac{\theta}{2}, \quad \mathbf{v} = \left(\sin\frac{\theta}{2}\right)\hat{\mathbf{n}} \tag{2.30}$$

#### Translation and orientation in 3-Dimension

The position and orientation change, between two coordinate frames as shown in Fig. 2.5, can have different representations of orientation, which must be combined with translation representation, to create a tangible representation of relative pose.

The two most practical representations are :

— the quaternion vector pair : $\xi \equiv (\mathbf{t}, \mathring{q})$ where $\mathbf{t} \in \mathbb{R}^3$ is the Cartesian position of the frame's origin with respect to the reference coordinate frame, and $\mathring{q} \in \mathbb{Q}$ is the frame's orientation with respect to the reference frame.

— the $4 \times 4$ homogeneous transformation matrix : its derivation is similar to the 2D case of Eq.2.21 but extended to account for the z-dimension.

$$^{A}\tilde{\mathbf{p}} = \begin{pmatrix} ^{A}\mathbf{R}_B & \mathbf{t} \\ \mathbf{0}_{1x3} & 1 \end{pmatrix} {}^{B}\tilde{\mathbf{p}} = {}^{A}\mathbf{T}_B \, {}^{B}\tilde{\mathbf{p}} \tag{2.31}$$

The Cartesian translation vector between the origin of the coordinates frames is $\mathbf{t}$ and the change in orientation is represented by a $3 \times 3$ orthonormal submatrix $R$. In Eq.2.31, the vectors are expressed in homogenous form to get a $^{A}\mathbf{T}_B$, which is a $4 \times 4$ homogeneous transformation, so $\xi \equiv \mathbf{T}$.

### 2.1.3 Reference frames

So far it has been established the mathematical tools to relate different frames of reference. When working with mobile robotics, two frameworks are usually used [3], one fixed on the robot in analysis and the other fixed to the environment, which constitutes the frame of reference.

When analyzing the motion of mobile robots, specially in 6 degrees of freedom (DOF), it is convenient to define a series of coordinate frames indicated in Fig. 2.6.

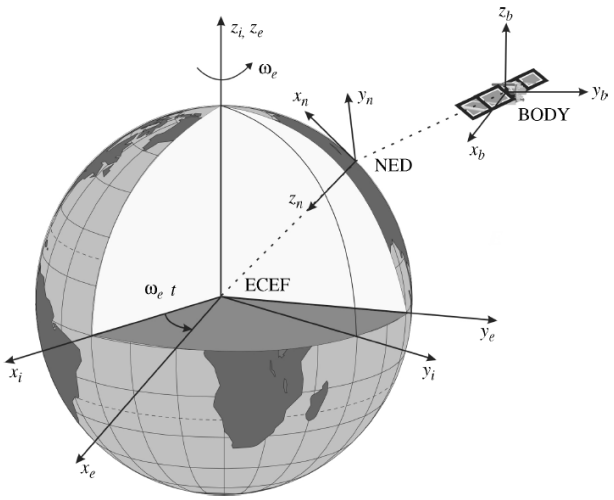



FIGURE 2.6 – Earth-centered Earth-fixed frame

— **The Earth-centered inertial frame (ECI)**
is an inertial frame for terrestrial navigation -i.e., a nonaccelerating reference frame in which Newton's laws of motion apply. The origin of the ECI coordinate frame $(x_i, y_i, z_i)$ is located at the center of the Earth with axes as shown in Fig.2.6.

— **The Earth-centered Earth-fixed reference frame (ECEF)**
$(x_e, y_e, z_e)$ has its origin fixed to the center of the Earth but the axes rotate relative to the inertial frame ECI which is fixed in space. The angular rate of rotation is $\omega_e = 7.2921 \cdot 10^{-5}$ rad/s. For marine vessels moving at relatively low speed, the Earth rotation can be neglected and hence the ECEF can be considered to be inertial. The ECEF is needed for global guidance, navigation and control, for example to describe the motion and location of mobiles in transit between different continents.

— **The North-East-Down coordinate system (NED)**
$(x_n, y_n, z_n)$ is defined relative to the Earth's reference ellipsoid (World Geodetic System 1984). This is the coordinate system we refer to in our everyday life. It is usually defined as the tangent plane on the surface of the Earth moving with the robot, but with axes pointing in different directions than the body-fixed axes of the vessel. For this system the x-axis points towards true North, the y-axis points towards East while the z-axis points downwards normal to the Earth's surface.

— **The body-fixed reference frame BODY**

$(x_b, y_b, z_b)$ is a moving coordinate frame which is fixed to the robot. The position and orientation of the robot are described relative to the inertial reference frame (approximated by the ECEF or NED) while the linear and angular velocities of the mobile should be expressed in the body-fixed coordinate system. The origin $0$ of the body-fixed frame is usually chosen to coincide with the center of gravity (CG) when CG is in the principal plane of symmetry, or at any other convenient point if this is not the case.

## 2.2 Modelling

Derivation of the model of a robot plays an important role for simulation of motion, analysis of structure, and design of control algorithms.

The fundamental principles of dynamics allow us to find the state equations of robotic systems. The calculation procedures of these models can be complicated for complex systems, so numerical approximations of them by CAD design are common.

### 2.2.1 Dynamic models

The most common methods to obtain the equations of motion in mechanical systems, in particular robots, are generally based on one of two formulations, the Lagrange formulation or the Newton-Euler formulation [1].

In *Lagrange formulation*, once a set of generalized coordinates is defined, from the Lagrangian of the mechanical system :

$$\mathcal{L} = \mathcal{T} - \mathcal{U} \tag{2.32}$$

where the expression of kinetic energy $\mathcal{T}$ and potential energy $\mathcal{U}$ of the system are considered, the set of motion equations is derived [4]. While the *Newton-Euler approach* is based on a balance of all the forces acting on the body of the robot, from where motion equations are obtained [5].

Based on the Newton-Euler approach, it is possible to summarize the obtaining procedure of the state equations of a system composed of several subsystems (parts) $\mathcal{S}_1$,$\mathcal{S}_1$,...,$\mathcal{S}_l$ , assumed rigid, following three steps [6] :

1. **Obtaining the differential equations :** For each subsystem $\mathcal{S}_k$, with mass $m$ and inertial matrix $J$, the following relations must be applied :

$$\sum_i \mathbf{f}_i = m\mathbf{a} \tag{2.33}$$

$$\sum_i \tau_{f_i} = J\dot{\omega} \tag{2.34}$$

   where the $\mathbf{f}_i$ are the forces acting on the subsystem $\mathcal{S}_k$, $\tau_{f_i}$ represents the torque created by the force $\mathbf{f}_i$ on $\mathcal{S}_k$, with respect to its CG. The vector $\mathbf{a}$ represents the tangential acceleration of $\mathcal{S}_k$ and the vector $\dot{\omega}$ represents the angular acceleration of $\mathcal{S}_k$. After decomposing these $2l$ vectorial equations according to their components, it is obtained $6l$ scalar differential equations such that some of them might be degenerate.
2. **Removing the components of the internal forces :** In differential equations there are the so-called bonding forces, which are internal to the whole mechanical system, even though they are external to each subsystem composing it. They represent the action of a subsystem $\mathcal{S}_k$ on another subsystem $\mathcal{S}_j$. Following the action–reaction principle, the existence of such a force, denoted by $\mathbf{f}^{k,j}$, implies the existence of another force $\mathbf{f}^{j,k}$, representing the action of $\mathcal{S}_j$ on $\mathcal{S}_k$, such that $\mathbf{f}^{k,j} = -\mathbf{f}^{j,k}$. Through a formal manipulation of the differential equations and by taking into account the equations due to the action-reaction principle, it is possible to remove the internal forces. The resulting number of differential equations has to be reduced to the number $n$ of degrees of freedom $q_1$,.., $q_n$ of the system.
3. **Obtaining the state equations :** We then have to isolate the second derivative $\ddot{q_1}$,...,$\ddot{q_n}$ from the set of $n$ differential equations in such a manner to obtain a vectorial relation such as :

$$\ddot{\mathbf{q}} = \mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{u}) \tag{2.35}$$

   where $\mathbf{u}$ is the vector of external forces that are not derived from a potential.

A mechanical system whose dynamics can be described by the relation $\ddot{\mathbf{q}} = \mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{u})$ will be referred to as *holonomic* (this implies that $f$ can be integrated). For a holonomic system, $\mathbf{q}$ and $\dot{\mathbf{q}}$ are thus independent. It means, that an holonomic system has the number of controllable degrees of freedom equal to the total degrees of freedom. When the system is restricted to kinematic coinstraints of the form $h(\mathbf{q}, \dot{\mathbf{q}}) = 0$, the system will be referred to as *non-holonomic*.

### 2.2.2 Kinematic models

Robotic systems are controlled by forces and torques, responding to dynamic models. These models are usually composed by coefficients that are difficult to estimate and usually not completely known. The same mechanical systems can be represented by kinematic models, where the control variables can be positions, speeds or accelerations. These models start from the premise that the control speed is reached instantaneously and without saturation, that is to say that the actuators (motors, pistons, etc.) are ideal and have some type of high-gain feedback that guarantees the commanded values. This closed loop system can be thought through the following control law [6] :

$$\mathbf{u} = k(\mathbf{x} - \mathbf{w}) \tag{2.36}$$

where $\mathbf{u}$ is the control signal applied to the actuator, $\mathbf{w}$ is the setpoint, and $\mathbf{x}$ the measured variable (position, speed or acceleration). Ideally, if the loop gain $k$ is very large, we can conclude that $\mathbf{x} \simeq \mathbf{w}$. In reality this is not always possible. This ideality could be maintained in a certain operating range, but in general it is not valid due to power reasons or actuator saturation. For this reason, the use of kinematic models is reserved for simple applications, while those requiring more precise control are usually implemented as superior layer of dynamic control-loops that ensure the assumptions of the kinematic model.

## 2.3 Work environment

Models of the environment are indispensable for the development of several applications in robotics systems. It is through these environment models that the robot can adapt its decisions to the current state of the world surrounding.

The bibliography shows a tendency to classify the modeling of the environment in those that are for indoors situations and those for outdoors [4]. On one hand, indoor environments are highly structured, they contain primarily linear structures such as lines and planes, so several of the modeling techniques just consider two-dimensions maps. On the other hand, outdoors environments are constituted by uneven surfaces, they include mobile elements and they are time variant.

Another way to classify work environments, even more general than the previously

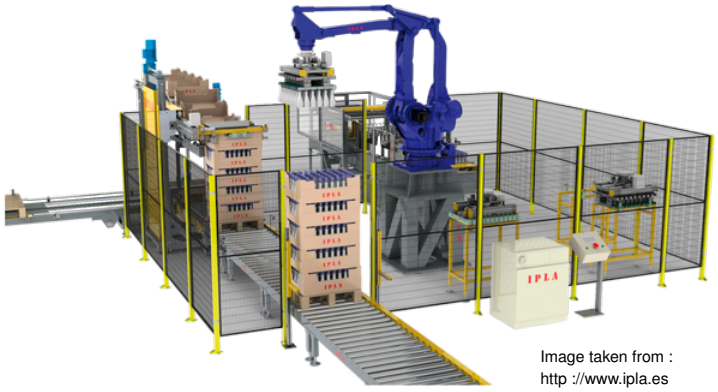



FIGURE 2.7 – Structured work environment : robotic cell

mentioned, is in structured and unstructured ones :

- **structured work environments**, the environment is perfectly known and if there are mobile elements their dynamics are known. This type of environment is common in industrial applications where robots are locked in work cells protected by physical or virtual barriers, see Fig. 2.7. If they are crossed, protocols of safe detention are activated [7].
- **unstructured environments**, in extreme cases there is no information about the environment and the robot only has the sensors information to generate the model of its environment. In other situations the robots have a partial map of the surroundings but it must share them with mobile elements. These work conditions force the use of complementary reactive techniques to avoid collisions. An example of this last case can be an indoor environment, where the structure can be known partially but the space is shared with people and other mobile elements, see Fig.2.8.

A large part of the environment modeling techniques are based on occupancy grids [8], line maps [9], and landmark based maps [10]. These are mainly used in two-dimensional environments with low dynamics. While other more complex technologies, used outdoors, are based on elevation grids, point sets or meshes which can consider heights, mobile elements and time variant maps [11][12]. Some important features for the work environment modeling are :

- ability to consider moving objects.
- achievable resolution, considering the on-board sensors of the robot in particular.


FIGURE 2.8 – Unstructured work environment : semi-automatic warehouse of postal company DHL

— power processing needed to be used.
— scalability for large areas of work.
— adaptation to uncertainty in robot and sensing positions.

## 2.4 Control objectives in mobile robotics

As it was presented in Chapter 1, the use of robots includes multiple activities such that welding, transport, inspection, mapping, etc. Most of these activities, in a particular work environment (see section 2.3), are supported on a tracking function. This function consists in following a path or trajectory, usually generated by a navigation or planning stage embedded on the robot itself.

The accuracy with which the robot can perform this task directly affects the main operation required. For example, during an inspection task the oscillating movement of the robot will reduce the quality of the images captured by its camera. Similarly, in an industrial follower the error in the track could lead to a drop in the quality of the final product or even accidents.

These activities, being inserted in the physical world, even in controlled environments, are prone to be interrupted by other agents of the environment, which will be called obstacles. Given this, it will be necessary to provide robots with techniques to avoid obstacles.

Taking into consideration the above, tracking and obstacle avoidance are between the main control objectives in robotics. In the following, details of both objectives will be presented as well as some approaches to their partial solution provided in the bibliography.

### 2.4.1 Tracking task

The tracking task is the basis of many of the advanced functions of both mobile and fixed robots. This task can be defined as following a series of points that make up a path or trajectory. In this point it is important to make the difference between path and trajectory :

1. The concept of *path* is understood here as a series of points in space (continuous or not) to be sequentially traversed, where the robot must have a particular pose. It means in space that leads from an initial pose to a final pose.
2. A *trajectory* is a path with a time stamp associated to each point of the path to travel (i.e. a spatial and temporal coordinate for each point of the route). For example there is a path from $A$ to $B$, but there is a trajectory from $A$ to $B$ in $10s$ or at $2\frac{m}{s}$.

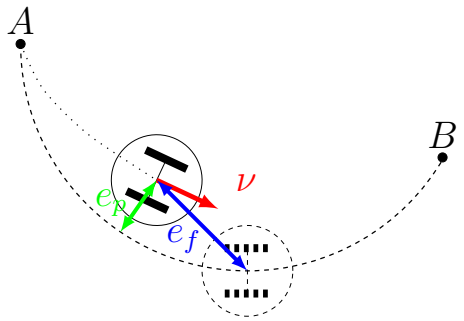



FIGURE 2.9 – Possible definitions of tracking errors

The tracking problem consists in following a reference generated by a higher control stage (motion planner), with the minimum possible error.

The definition of the error is based on the distance between the current position of the robot and the reference position.

In the case of trajectory tracking, the reference point moves according to the speed profile or time stamp for each point of the trajectory. Whereas in the case of path

following, the reference position will depend on the technique used. In the bibliography we can find two approaches :

1. Techniques that establish the point of reference as the point of the path closest to the robot [13][14].
2. Techniques in which the control algorithm generates a virtual target which must be followed by the robot [15][16]. One of the particularities of this approach is the possibility of parametrizing the reference [17].

Fig.2.9 shows a graphic interpretation of the error for both cases. Here $e_p$ represents the error to the closest point of the path while $e_f$ can be interpreted as the error of the trajectory following or the path following with a virtual target. We can add that the second approach can be interpreted as an improvement as it avoids the singularities that occur when the vehicle is located at the center of curvature of the path (where the closest point is not unique), allowing for global convergence of the vehicle to the desired path [18].

From the previous, the problem of the path generation and its continuity arises. Depending on the particular application, the path can be a sequence of points in space from which a control system generates the corresponding signals to converge to them following the given reference, see Fig. 2.10. While others, for example in manipulators, require continuous values generated by the planning stage generating a smooth path. Smoothness in this context means that its first two temporal derivatives (speed and acceleration) are continuous and sometimes also the derivative of acceleration or jerk. One of the techniques to obtain smooth paths from a discrete sequence of points is to use quintic (fifth-order) polynomials [2].

To obtain successful performance of the solution to the tracking problem, significant effort has been devoted to the development of model-based control strategies [19][20]. Among the control approaches reported in the literature, typical methods include inverse dynamic control, feedback linearization, and passivity based control [4]. In the present work, proposals for path tracking techniques with virtual reference will be addressed in chapters 3 and 4.

### 2.4.2 Obstacle avoidance

A *global path planning* algorithm generally uses a priori information to build a complete model of the surrounding environment and then try to find the best possible so-

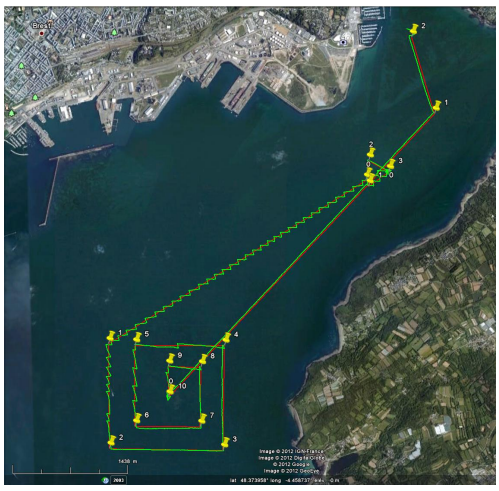


FIGURE 2.10 – Discrete path following for a marine vessel

lution. But in unknown or unstructured environments, this is not sufficient so it is necessary to combine the path planning method with a *local or reactive navigation* using on board sensors, so as to locally observe small fragments of the surrounding at each time. In such scenario, the problem of detection and reaction in the presence of obstacles arises.

In the case of mobile robots the most common approaches are : first, to use a belt of proximity sensors (ultrasound, infrared, sonar, ...) mounted on the vehicle, allowing a discrete scanning of the space around the robot [21] ; and in second place, the use of a rotating laser beam, frequently coupled with a vision system, resulting in continuous estimation of the free region around the vehicle [22].

Once the necessary information is obtained about the surrounding environment, the optimal way to process and take action will depend on the particular situation where our system will be.

Among the most extended methods it is possible to mention : Potential Fields [23], Vector Field Histogram [24] and Velocity obstacles methods [25]. These methods modify the robot pose (position and orientation) when a collision situation arises, indeed they are designed for environments where the robot is allowed to leave the pre-elaborated path.

In the path following operations a time variant target $\eta$ moves through the path, while a low control level (dynamic or kinematic) reduces the tracking error. Then, the objective is to generate a collision-free variation of the target $\eta$ so that the robot follows the path in a safe way, respecting a $d_{safe} > 0$ distance to the obstacles in the environment, see Fig. 2.11.

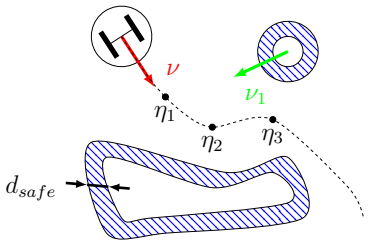



FIGURE 2.11 – Path following with obstacle avoidance

## 2.5 Problems on the control of mobile robots

The fields of robotic application are very varied, so are the control problems that arise from their utilization. In particular, since this work approaches robotics from the control theory, two major problems can be distinguished : constraints and non-idealities. Constraints refer to all types of limitations present in the robot, while non-idealities are related to unknown elements of the robot and its environment which affect its performance.

Regarding the constraints, it is possible to classify them as :

— **input constraints :** These can be understood as physical limitations on the actuator elements. These limitations can be generated by dead zones and actuator saturations which limit maximum forces or speeds. Among the most common consequences, we can find the incorrect following of a reference (tracking applications), and in the case of linearized systems, non-compliance with modeling conditions. Furthermore, the use of techniques that do not take into consideration these restrictions may lead to unused capacities as well as premature deterioration of the involved hardware.

— **output constraints :** are understood as limitations of the work environment. These can be fixed or dynamic. Study cases can included industrial manipulators, which commonly share their workspace with human personnel limiting it due to safety issues. Also, the obstacle avoidance problem can be considered as a problem of output constraint, as possible positions of robot are limited by the presence of other components of the environment.

— **structural constraints :**
This type of limitations refers to restrictions inherent to the hardware or software embedded on the robot. It is common to find limitations in terms of the

control that it is possible to configure, to have limited sensors (number of them, maximum range or sampling frequency, etc), or even mechanical restrictions for example in mobile robots the non-holonomic robot kinematic restrictions.

Non-idealities refer to elements present in robots that hinder their modeling or control, especially those elements that are not completely modeled. It is possible to group them into three groups :

— **uncertainties :** The control of systems, especially robotic systems, requires a modeling of the intervening physical processes (see section 2.2). This modeling, by simplification, often discards part of the physical behavior of these systems, or approximates the parameters involved. These approximations, usually of varying parameters, lead to models which not behave exactly like the real system, conjecturing the need to implemented robust controls against possible parametric variations. Hydrodynamic parameters in marine vehicles or variant payloads in transport applications are examples of this type of uncertainties.

— **disturbances :** The disturbances refer to elements of the environment that are not totally modeled which affect the robot operation (see section 2.3). For example, winds in aerial applications, marine currents in AUV or ground variations in terrestrial applications. It is expected that the behavior of the control can reject these disturbances in a quick and stable way, so robust control techniques are also required.

— **non-linearities :** These can be understood as a control problem or as a characteristic of the robotic systems. Although a correctly modeled non-linearity does not present a problem by itself, its difficulty is associated with the problem of robust control design without falling into a linearization, which is only valid for a given operating point. Most of the robust control techniques found in bibliography need for a linearized model of the system.

In the following chapters of this work, we will address partial solutions to the presented problems.

CHAPITRE 3

# INPUT-CONSTRAINED ROBOTIC CONTROL

This chapter analyzes the problem of systems with restricted inputs, in particular an autonomous underwater vehicle (AUV) case is studied. The constraints analyzed can be real or virtual and they obey to different criteria such as energy, system security, or simply physical limitations of the plant. Here, it is addressed both the modeling and control of the experimental Ciscrea AUV. A six degree-of-freedom model is presented and validated for turn and emerge/sink maneuvers. Then, a constraint compensating algorithm is proposed based on quasi sliding mode conditioning ideas, and added to a pre-existing PD controller in order to improve the overall closed-loop response. By considering actuator constraints, the employed technique allows path following at greater speed than the original controller for a given error tolerance. Experimental results on the so-called Ciscrea underwater robot are presented.

## 3.1 Case study : AUV Ciscrea control

The study of the marine environment and commercial activities offshore usually has high cost due to the infrastructure necessary, equipment and skilled personnel. The relevant campaigns in the Arctic and Antarctica [26], the study of seabed [27], the research in algal blooms and the analysis of the stock in fisheries surveys [28], applications in the oil and gas sector [29], among others, prove that the oceans can be successfully explored with robotic probes. The use of autonomous robots for these activities, especially AUV, has powered this kind of research.

During these activities, AUVs are exposed to an unknown environment where tasks like infrastructure inspection, patrolling areas, or carrying elements are the most common. These duties share a common goal : to follow a pre-established path, as fast as possible and with the minimum possible error.

As can be seen, a performance trade-off arises. Indeed, if a sharp path is asked

for the guidance function of the robot or if a very quick answer is required for the path completion, the actuators will reach the maximum allowed allocation. The saturation phenomenon will occur and either error or speed performance metrics will degrade (when saturation occurs, the system behaves as in open-loop manner). Thus, physical constraints should be taken into account as long as demanding control objectives are required.

Bibliography presents several cases of study about the saturation problem in autonomous systems [30]. In particular for marine systems, several related works can be found in the literature and we present here some of them. In the works by Campos et al.[31] a PD nonlinear control based on saturation functions with varying parameter for depth and yaw set-point regulation and trajectory tracking on an underwater vehicle is proposed. Zheng et al.[32] deal with asymmetric saturation over the actuator of a marine vessel. In their work, a Gaussian error function-based continuous differentiable asymmetric model is employed, for the design of the base control with backstepping technique. Another attempt to AUV application is presented by Steenson et al.[33] where the saturation of the actuator was considered directly in the controller tuning through a MPC design. Moreover, Sarhadi et al.[34] take a simpler approach through a model reference adaptive controller with an anti-windup action, which acts over the input signals of an AUV when saturation arises.

These last solutions are valuable and they achieve good results, but in general they require a good model of the system and actuators. In other fields of robotics control techniques based on time-delay estimations were used to overcome with these issues [35], [36], [37], but they do not take into account the constraint problematic over the actuators.

Considering the path following applications, the path to follow is frequently specified as a vector input function which may be parameterized in terms of a motion parameter, as it is proposed in the works of Nenchev[38], [17] and Garelli[16]. Both actuator limits and error tolerances give rise to a tracking speed limit at each point of the path. Traditional control strategies (particularly in commercial robots) have tackled this problem by using a conservative constant tracking speed so that actuators never reach their limits, or they reach them only in some isolated point of the path. A better solution, naturally, requires using variable tracking speed. However, this is in general computationally awkward as the maximum tracking speed compatible with the error tolerance must be computed on-line as the robot advances through the path.

In this chapter, a simple algorithm for variable speed tracking computation considering actuator limits is developed and experimentally tested. This is based on quasi sliding mode ideas originally proposed by Garelli et al. [16, 39] for kinematic control frameworks. Unlike these first works, here a methodology is developed to be applied to dynamic frameworks and, additionally, it is experimentally validated.

For illustrative purposes, the method is evaluated in the experimental AUV Ciscrea robot under a closed-controller restriction, i.e. the original AUV controller is fixed and inaccessible due to software and safety constraints. The Ciscrea robot is shown in Fig. 3.1, and its main characteristics can be seen in Table 3.1. Due to its difficult hydrodynamic modeling and identification, this robot has been used to test different control laws explained in [40].

A detailed modeling, together with its validation, of the AUV Ciscrea is addressed in section 3.2. Then, section 3.3 presents the details of the constraint compensation technique, while section 3.4 is devoted to experimental results.

FIGURE 3.1 – The AUV Ciscrea

## 3.2 Underwater robot model : AUV Ciscrea

In this section a dynamic and kinematic control-oriented model for the AUV under study will be developed following the Newton-Euler approach of subsection 2.2.1, together with ideas proposed in the books of Fossen [3] and Antonelli [41]. In addition, the modelling is validated over the AUV Ciscrea that is available at ENSTA Bretagne.

TABLE 3.1 – Ciscrea main characteristics

| | |
|---|---|
| Size (L, W, H) | (0.525m, 0.406m, 0.395m) |
| Weight | 15.56kg |
| Actuation | 6 thrusters (2 are vertical and 4 are horizontal) |
| Speed Range | 0-2 knots for surge and 0-1 knot for sway and heave |
| Max Depth | 50m |
| Depth sensor | Range $0 - 100$m<br>Absolute precision $+/-10$cm |
| Yaw sensor | Resolution $0.1°$<br>Precision $+/-1.5°$ |

### 3.2.1 Model description

Two coordinate systems are commonly employed for localization (refer to subsection 2.1.3), as can be seen in Fig. 3.2 :

— the classical earth frame, called NED-frame due to the main directions that are North East and Down ;

— the frame linked to the robot, called B-frame due to the Body fixed reference

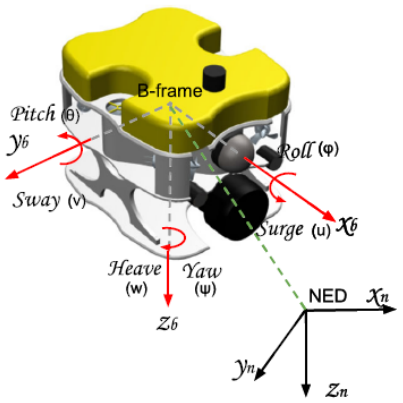



FIGURE 3.2 – Ciscrea's coordinate systems

All the data are given in the international units : distances are in meters, angles in radians and positive clockwise. The position, velocity and force are denoted $\eta$, $\nu$ and $\tau$.

They are defined as follows :

$$\begin{aligned} \boldsymbol{\eta} &= [x, y, z, \phi, \theta, \psi]^T \text{(position)} \\ \boldsymbol{\nu} &= [u, v, w, p, q, r]^T \text{(velocity)} \\ \boldsymbol{\tau} &= [X, Y, Z, K, M, N]^T \text{(force and torque)} \end{aligned} \tag{3.1}$$

while the rigid-body dynamics is given by :

$$M_{RB}\dot{\boldsymbol{\nu}} + C_{RB}(\boldsymbol{\nu})\boldsymbol{\nu} = \boldsymbol{\tau_{env}} + \boldsymbol{\tau_{hydro}} + \boldsymbol{\tau_{pro}} \tag{3.2}$$

and the hydrodynamic formulation (hydrostatics included) is :

$$\boldsymbol{\tau_{hydro}} = -M_A\dot{\boldsymbol{\nu}} - C_A(\boldsymbol{\nu})\boldsymbol{\nu} - D(|\boldsymbol{\nu}|)\boldsymbol{\nu} - \boldsymbol{g(\eta)} \tag{3.3}$$

The corresponding parameters are given in Table 3.2.

TABLE 3.2 – Nomenclature of the underwater vehicle model

| Parameter name | Definition |
|---|---|
| $M_{RB}$ | Mass and inertia matrix for the rigid model of the robot |
| $M_A$ | Added mass matrix used for marine vehicles |
| $C_{RB}$ | Rigid-body matrix induced by Coriolis phenomenon |
| $C_A$ | Added mass matrix induced by Coriolis phenomenon |
| $D(\lvert\boldsymbol{\nu}\rvert)$ | Damping matrix due to mechanical frictions |
| $\boldsymbol{g(\eta)}$ | Restoring forces and moments vector |
| $\boldsymbol{\tau_{env}}$ | Disturbances from environment (wind,waves and currents, even if when deep the wind end waves can be neglected) |
| $\boldsymbol{\tau_{hydro}}$ | Vector of the hydrodynamic forces and moments |
| $\boldsymbol{\tau_{pro}}$ | Propeller forces and moments vector |

In the present application, as the Ciscrea vehicle speed remains low, the parameters $C_{RB}$ and $C_A$ can be neglected, while the parameters $M_{RB}$, $M_A$, and the damping

matrix are obtained from Yang et al.works [42, 43].

The vector $\boldsymbol{g}(\boldsymbol{\eta})$ that represents the forces and the moments produced by the weight and the buoyancy forces applied to the rigid-body is given by the following formulation :

$$\boldsymbol{g}(\boldsymbol{\eta}) = \begin{bmatrix} -(m - \rho\ vol)g\sin\theta \\ (m - \rho\ vol)g\cos\theta\sin\phi \\ (m - \rho\ vol)g\cos\theta\cos\phi \\ -BG_y mg\cos\theta\cos\phi + BG_z mg\cos\theta\sin\phi \\ -BG_z mg\sin\theta + BG_x mg\cos\theta\sin\phi \\ -BG_x mg\cos\theta\sin\phi - BG_y mg\sin\theta \end{bmatrix} \tag{3.4}$$

where :

- the vector $\boldsymbol{BG} = [BG_x, BG_y, BG_z]^T$ represents the vector from the center of gravity, denoted CG, to the buoyancy center, denoted CB,
- the scalar $\rho$ is the fluid density which can vary according to the sea or lake and environment temperature,
- the scalar $vol$ is the displaced fluid volume which determineS the Archimede force,
- the scalar $g$ is the gravity acceleration module,
- $m$ is the AUV mass.

The convention for the angle measure is the one shown in Fig. 3.2.

It is important to notice that the center of buoyancy, CB, and the center of gravity, CG, were experimentally adjusted by moving, adding or removing payload and floats. Both are actually close and they can be considered in the same location. Furthermore, due to the symmetric form of the AUV, CB and CG coincide with the geometrical center of the robot.

The marine disturbances are mainly due to wind, waves and current. They contribute to $\boldsymbol{\tau_{env}}$ term. However, as the vehicle is most of the time under the water surface, waves and wind are not strong enough to have a real effect on the robot. Then only marine currents are considered during underwater operations.

Concerning the hydrodynamics parameters, it is worth mentioning :

- the term $M_A \in \Re^{6\times6}$ is the added mass. This is a classical virtual concept used in marine mechanics for representing the hydrodynamic forces and moments. Indeed any object in a fluid will encounter this $M_A$ as soon as it has an accele-

ration. This is due to the important inertia of the fluid (in the air, the low density makes this phenomenon negligible in comparison with the other forces).

— the term $D(|\boldsymbol{\nu}|) \in \Re^{6\times6}$ represents the so-called fluid damping. It can be decomposed in four parts :
  — potential damping,
  — wave drift damping,
  — skin friction,
  — vortex shedding damping.

  As explained by Yang R. [42], the first two could be dismissed in this kind of application, and the two others are given by approximations. In order to be precise enough, a second order approximation has been chosen and then the term $D(|\boldsymbol{\nu}|)$ could be given by linear and quadratic matrices, $D_L$ and $D_N$ respectively, as shown in the following equation :

$$D(|\boldsymbol{\nu}|) = D_L + D_N|\boldsymbol{\nu}| \tag{3.5}$$

The kinematic relation of velocity vector $\boldsymbol{\nu}$ (in B-frame) and the position vector $\boldsymbol{\eta}$ (in NED-frame) can be write as :

$$\dot{\boldsymbol{\eta}} = J(\boldsymbol{\Theta})\boldsymbol{\nu} \tag{3.6}$$

where $J(\boldsymbol{\Theta}) \in \Re^{6\times6}$ is a transformation matrix between B-frame and NED-frame defined in Equations 3.7, 3.9 and 3.8 with $\boldsymbol{\Theta} = [\phi, \theta, \psi]^T$ (This transformation matrix $J$ is obtained in the same way as it has been done in section 2.1 for two and three dimensions).

$$J(\boldsymbol{\Theta}) = \begin{bmatrix} R(\boldsymbol{\Theta}) & 0_{3x3} \\ 0_{3x3} & T(\boldsymbol{\Theta}) \end{bmatrix} \tag{3.7}$$

$$R(\boldsymbol{\Theta}) = \begin{bmatrix} \cos(\psi)\cos(\theta) & -\sin(\psi)\cos(\phi) + \cos(\psi)\sin(\theta)\sin(\phi) & \sin(\psi)\sin(\phi) + \cos(\psi)\cos(\phi)\sin(\theta) \\ \sin(\psi)\cos(\theta) & \cos(\psi)\cos(\phi) + \sin(\phi)\sin(\theta)\sin(\psi) & -\cos(\psi)\sin(\phi) + \sin(\theta)\sin(\psi)\cos(\phi) \\ -\sin(\theta) & \cos(\theta)\sin(\phi) & \cos(\theta)\cos(\phi) \end{bmatrix} \tag{3.8}$$

$$T(\boldsymbol{\Theta}) = \begin{bmatrix} 1 & \sin(\phi)\tan(\theta) & \cos(\phi)\tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \frac{\sin(\phi)}{\cos(\theta)} & \frac{\cos(\phi)}{\cos(\theta}) \end{bmatrix} \tag{3.9}$$

From B-frame model to NED-frame model, the transformation is possible applying $J(\mathbf{\Theta})$ to Eq. 3.2 and 3.3 in order to obtain the differential equation which describes the robot behavior :

$$M^*\ddot{\boldsymbol{\eta}} + D^*(|\boldsymbol{\nu}|)(\dot{\boldsymbol{\eta}}) + \boldsymbol{g^*(\eta)} = \boldsymbol{\tau_{pro}} + \boldsymbol{\tau_{env}} \tag{3.10}$$

with the following new notations :

— $M^* = J^{-T}(\mathbf{\Theta})(M_{RB} + M_A)J^{-1}(\mathbf{\Theta})$, equivalent to the mass,
— $D^*(|\boldsymbol{\nu}|) = J^{-T}(\mathbf{\Theta})D(|\boldsymbol{\nu}|)J^{-1}(\mathbf{\Theta})$, equivalent to the damping term,
— $\boldsymbol{g^*(\eta)} = J^{-T}(\mathbf{\Theta})\boldsymbol{g(\eta)}$, equivalent to forces and moments.

According to these equations, the control-oriented model can be represented with the Eq. 3.11. This model will be used for intensive simulations in order to show the results in various configuration of the robot and with varying environments.

$$\left\{ M_A + \begin{pmatrix} m & 0 & 0 & 0 & 0 & 0 \\ 0 & m & 0 & 0 & 0 & 0 \\ 0 & 0 & m & 0 & 0 & 0 \\ 0 & 0 & 0 & I_x & 0 & 0 \\ 0 & 0 & 0 & 0 & I_j & 0 \\ 0 & 0 & 0 & 0 & 0 & I_k \end{pmatrix} \right\} \begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \\ \dot{p} \\ \dot{q} \\ \dot{r} \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{n} [T_i \cos(\psi_{t_i}) \cos(\theta_{t_i})] - (m - \rho v_{ol}) g \sin(\theta) - D_{Nu}|u|u + D_{Lu}u + m(rv - qw) \\ \sum_{i=1}^{n} [T_i \sin(\psi_{t_i}) \cos(\theta_{t_i})] + (m - \rho v_{ol}) g \cos(\theta) \sin(\phi) + D_{Nv}|v|v + D_{Lv}v + m(pw - ru) \\ -\sum_{i=1}^{n} [T_i \sin(\theta_{t_i})] + (m - \rho v_{ol}) g \cos(\theta) \cos(\phi) - D_{Nw}|w|w + D_{Lw}w + m(qu - pv) \\ -\sum_{i=1}^{n} \left[T_i \left(y_{t_i} \sin(\theta_{t_i}) + z_{t_i} \sin(\psi_{t_i}) \cos(\theta_{t_i})\right)\right] - D_{Np}|p|p + D_{Lp}p + (I_y - I_z)qr \\ \sum_{i=1}^{n} \left[T_i \left(z_{t_i} \cos(\psi_{t_i}) \cos(\theta_{t_i}) + x_{t_i} \sin(\theta_{t_i})\right)\right] - D_{Nq}|q|q + D_{Lq}q + (I_z - I_x)rp \\ \sum_{i=1}^{n} \left[T_i \left(x_{t_i} \sin(\psi_{t_i}) \cos(\theta_{t_i}) - y_{t_i} \cos(\psi_{t_i}) \cos(\theta_{t_i})\right)\right] - D_{Nr}|r|r + D_{Lr}r + (I_x - I_y)pq \end{pmatrix} \tag{3.11}$$

In Eq. 3.11, transformations from each propeller frame to the B-frame and interactions due to the angular momentum are considered, and with the following assumptions :

— $T_i$ is the torque generated by the thruster #$i$ of the robot for which $x_i$, $y_i$, $z_i$, $\phi_{t_i}$, $\theta_{t_i}$, $\psi_{t_i}$ denote respectively the surge, sway, heave, roll, pitch and heading of the thruster-frame with respect to B-frame, using the convention of Fig. 3.2. For the numerical values see Table 3.3.
— $D_{Ni}$ and $D_{Li}$ are the non-linear and linear damping coefficients for $i$-direction.

TABLE 3.3 – Thruster's parameters

| Thruster_i | | $x_i$ | $y_i$ [m] | $z_i$ [m] | $\psi_i$ [rad] | $\theta_i$ [rad] | $\phi_i$ [rad] |
|---|---|---|---|---|---|---|---|
| Horizontal thrusters | 1 | 0.165 | 0.145 | -0.05 | -0.5281 | 0 | 0 |
| | 2 | 0.165 | -0.145 | -0.05 | 0.5281 | 0 | 0 |
| | 3 | -0.165 | 0.145 | -0.05 | 3.6697 | 0 | 0 |
| | 4 | -0.165 | -0.145 | -0.05 | 2.6135 | 0 | 0 |
| Vertical thrusters | 5 | 0 | -0.14 | -0.05 | 0 | $\pi/2$ | 0 |
| | 6 | 0 | 0.14 | -0.05 | 0 | $\pi/2$ | 0 |

### 3.2.2 Model validation

The mechanical parameters were identified from laboratory measures on the experimental robot and from data provided by the manufacturer, whereas the hydrodynamic parameters were taken from Yang et al.[42].

Two additional effects were considered for a realistic description of the robot :

— A depth sensor delay that was estimated by experiments to be 0.5 s.

— The non-linear relation between the commanded digital torque signal $T_d$ (-127 to 127) to the real torque $T_a$ in each motor (Fig. 3.3). The conversion function was synthesized using linear regression from measures realized over the robot. The final expression of the conversion can be expressed as Eq. 3.12 and a graphic representation is made in Fig. 3.4.

$$T_a = \begin{cases} 4.7 & if \quad T_d \geq 127 \\ 3.2 \max\left(\frac{T_d}{203.874}, \frac{T_d - 30.3781}{65.6756}\right) & if \quad 0 < T_d < 127 \\ -4.3 \max\left(\frac{T_d}{203.874}, \frac{T_d - 30.3781}{65.6756}\right) & if \quad -127 < T_d < 0 \\ -6.32 & if \quad T_d \leq -127 \end{cases} \tag{3.12}$$

For validation of the proposed model, a comparison between simulations and experimental tests in a pool was performed. The comparison was done with the logged open-loop time responses. In Fig. 3.5 and Fig. 3.6, it is possible to appreciate the comparison of the yaw direction, during turning maneuvers, between the simulator and

FIGURE 3.3 – Ciscrea's Thruster

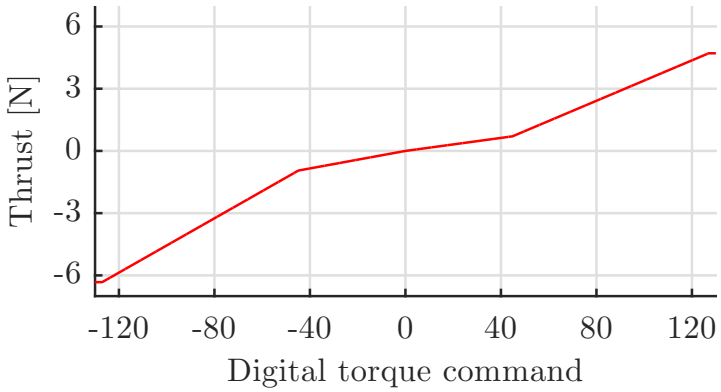



FIGURE 3.4 – Link between the digital control and the applied thrust

angle's measures in a real robot. In both cases a command torque signal is first sent to the robot's motors and then the free response of the system is observed. In the same way, Fig. 3.7 and Fig. 3.8 compare the depth direction for emerge and sink maneuvering, but in these cases the torque command is sustained over time due to the slow dynamics of the system.

To complete the evaluation of the model given, Table 3.4 presents different error calculus of the presented comparisons. Through the classical analysis of the Root-Mean-Square Error (RMSE), it is possible to show that the model is more accurate in the heave direction than in yaw. Nevertheless, considering the Normalized Mean Absolute Error (NMAE) it is observed that the weight of the errors in consideration in all the cases is smaller than 10%. To conclude, the Bias is obtained, where it can be noted that the model is able to predict the robot dynamics in an acceptable way for a control oriented simulator.

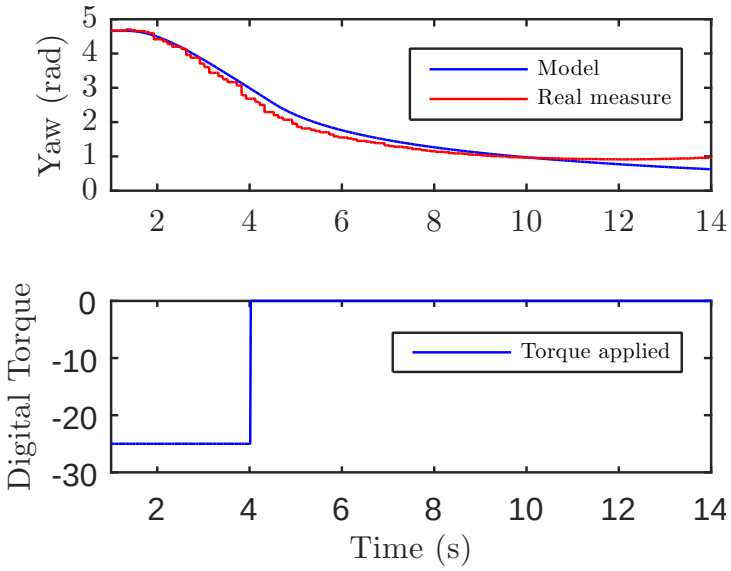



FIGURE 3.5 – Turning left maneuvering comparison

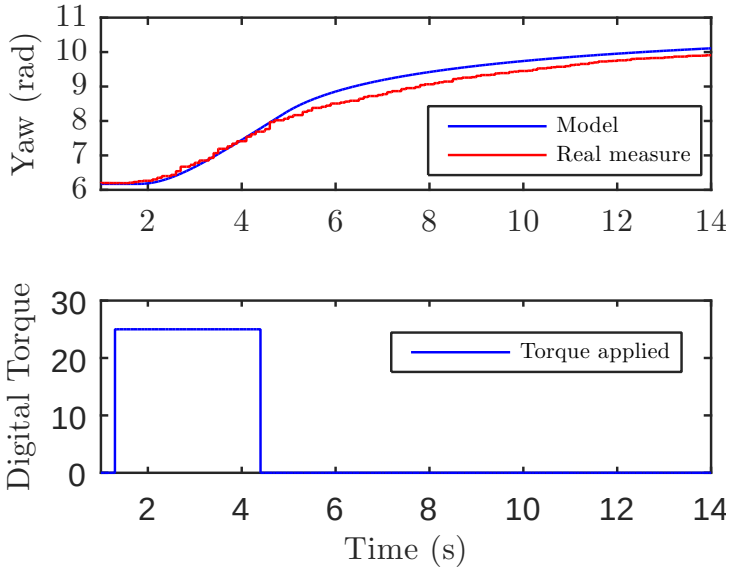



FIGURE 3.6 – Turning right maneuvering comparison

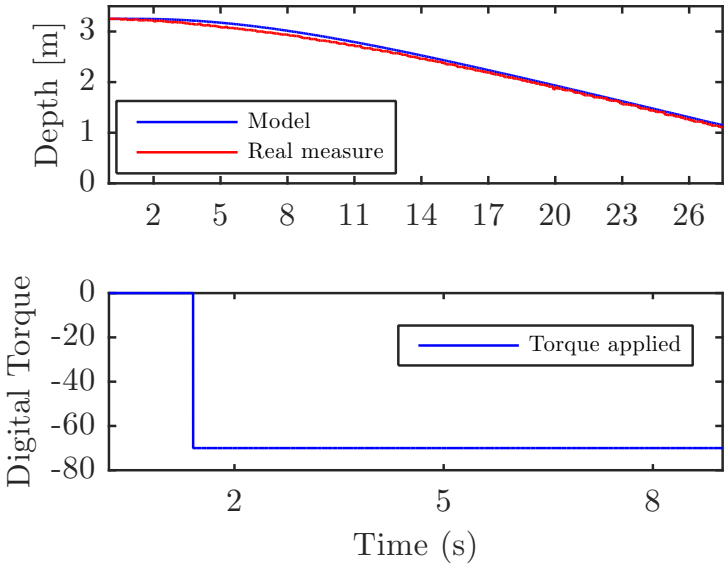



FIGURE 3.7 – Emerge maneuvering comparison

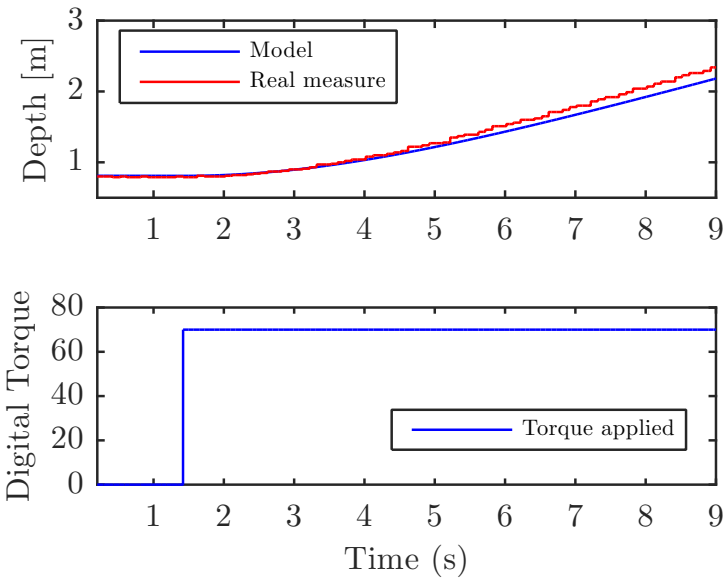



FIGURE 3.8 – Sink maneuvering comparison

TABLE 3.4 – Model error calculations

| Maneuver | RMSE | NMAE | Bias |
|---|---|---|---|
| Turning right | 0.2503 | 0.0247 | 0.1920 |
| Turning left | 0.1539 | 0.0802 | 0.0852 |
| Sink | 0.0805 | 0.0382 | -0.0537 |
| Emerge | 0.0587 | 0.0238 | 0.0545 |

## 3.3 Quasi sliding-mode conditioning

In this part, the basics of Sliding Mode (SM) control are revisited in order to recall its main ideas. Then a quasi sliding-mode algorithm is presented to deal with constraints. The objective of this point is to apply this method to the path following of the underwater robot presented before to attenuate the effects of input constraints.

### 3.3.1 SM Background

As explained by Utkin et al.[44], the classical sliding modes were developed for dynamic systems given by ODE (Ordinary Differential Equations). For these systems, it is possible to impose a robust closed-loop dynamics by applying a discontinuous control action.

In order to sum it up, a switching function is defined and according to its sign, the control signal can take one of two different values defined by a discontinuous control law with an associated manifold on the state-space (also called the sliding surface). This approach allows enforcing the system to have a response in two steps :

1. the first action is to reach the sliding surface
2. once the sliding surface is reached, the objective is to slide on it through a very fast switching control.

Once this particular mode of operation is established, known as sliding mode (SM), the prescribed manifold imposes the new and desired system dynamics.

In order to illustrate this behavior, let consider a nonlinear system of the form :

$$\begin{cases} \dot{\mathbf{x}} = f(\mathbf{x}) + g(\mathbf{x})u \\ y = h(\mathbf{x}) \end{cases} \tag{3.13}$$

with $\mathbf{x} \in \Re^n$ the state vector, $u$ the control action, $y$ the output and $f(\mathbf{x})$, $g(\mathbf{x})$, $h(\mathbf{x})$ vector fields. A discontinuous control law

$$u = \begin{cases} u^- & if \quad \sigma(\mathbf{x}) < \mathbf{0} \\ u^+ & if \quad \sigma(\mathbf{x}) > \mathbf{0} \end{cases} \tag{3.14}$$

is defined according to the sign of a switching function $\sigma(\mathbf{x})$. The sliding surface $S$ is defined as the manifold where the switching function vanishes. If the switching law in Eq. 3.14 fulfills the reaching condition

$$\begin{cases} \dot{\sigma}(\mathbf{x}) < \mathbf{0} & if \quad \sigma(\mathbf{x}) > \mathbf{0} \\ \dot{\sigma}(\mathbf{x}) > \mathbf{0} & if \quad \sigma(\mathbf{x}) < \mathbf{0} \end{cases} \tag{3.15}$$

locally near the surface (on both sides of it). The control is then switching at high frequency in order to constrain the trajectory of the state $x$ to slide on the surface $S$.

A necessary condition for satisfying Eq. 3.15 is that $\dot{\sigma}(\mathbf{x})$ explicitly depends on $u$, which is known as transversality condition [44].

### 3.3.2 SMRC (Sliding Mode Reference Conditioning)

The so-called Sliding Mode Reference Conditioning (SMRC) technique takes advantage from the high frequency switching of sliding regimes [45].

Differing from conventional SM, SMRC typically acts on one side of the surface, but it does nothing on the other side. So, it can be seen as a one-way SM. Also, as SMRC only becomes active when a constraint is reached (or about to be reached) but it turns off when that risk is over, it gives rise to transient quasi-SM operation on the limit surfaces (differing from conventional SM, in which after a reaching mode the desired operation is on the sliding surface).

Considering a constrained subsystem $S_c'$ of the closed-loop composed by the system in Eq. 3.13 and a given controller, an expanded system $S_c$ including $S_c'$ and a filter $F(s)$ (Eq. 3.24) is built (see Fig. 3.9), with the following state-space description :

$$S_c : \begin{cases} \dot{\mathbf{x}}_\mathbf{s} = f(\mathbf{x}_\mathbf{s}) + g(\mathbf{x}_\mathbf{s})w \\ v = h_v(\mathbf{x}_\mathbf{s}) \end{cases} \tag{3.16}$$

where the vector $\mathbf{x}_\mathbf{s}$ is the state vector and $v$ the constrained variable (which might represent the plant input, an internal state or even the controlled variable). In order to

specify the bounds on the variable $v$, the set $\mathbf{\Sigma}(\mathbf{x_s})$ is defined as follows :

$$\mathbf{\Sigma}(\mathbf{x_s}) = \{\mathbf{x_s} \mid \sigma(v) \leq 0\} \tag{3.17}$$

It is aimed to generate a control input $w$ which makes the system remain within $\mathbf{\Sigma}$. So, the right term of the first equation in Eq. 3.16 must be oriented to the interior of $\mathbf{\Sigma}$ at all points on the border :

$$\partial\mathbf{\Sigma} = \{\mathbf{x_s} \mid \mathbf{x_s} \wedge \sigma(v) = 0\} \tag{3.18}$$

which is achieved if [45] :

$$w = \begin{cases} \leq w^\sigma : \mathbf{x_s} \in \partial\mathbf{\Sigma} \wedge L_g\sigma > 0 \\ \geq w^\sigma : \mathbf{x_s} \in \partial\mathbf{\Sigma} \wedge L_g\sigma < 0 \\ \nexists \quad : \mathbf{x_s} \in \partial\mathbf{\Sigma} \wedge L_g\sigma = 0 \\ free : \mathbf{x_s} \in \mathbf{\Sigma} \setminus \partial\mathbf{\Sigma} \end{cases} \tag{3.19}$$

with $w^\sigma$ a scalar magnitude defined as :

$$w^\sigma = -L_f\sigma / L_g\sigma \tag{3.20}$$

The generic operator $L_q\zeta(x) : \Re^n \to \Re$ denotes the directional or Lie derivative :

$$L_q\zeta(x) = \frac{\partial\zeta}{\partial x}q(x). \tag{3.21}$$

which denotes the derivative of a scalar field $\zeta(x) : \Re^n \to \Re$ in the direction of a vector field $q(x) : \Re^n \to \Re^n$.

Note that the control action $w^\sigma$ is required to keep the system just in the neighbourhood of the invariant region border, $L_g\sigma \neq 0$ is the necessary transversality condition for SM to exist, and that $L_f\sigma > 0$ was assumed without loss of generality.

From Eq. 3.19 it can also be seen that $w$ might be freely chosen inside the region $\mathbf{\Sigma}$. Taking $w = 0$, which does not affect the original control system when constraints are not reached, it is possible thus make $\mathbf{\Sigma}$ invariant by implementing an auxiliary loop like the one of Fig. 3.9.

Additionally, a generic additive disturbance $d$ at the input of the constrained subsystem can be considered, such as $u = r_f + d$. Assuming $S_c'$ is a biproper dynamical

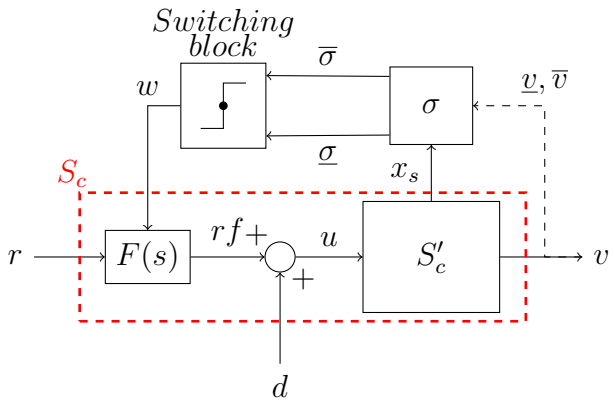



FIGURE 3.9 – Block diagram of SMRC technique.

system (i.e. with relative degree equal zero), the following switching law is therefore implemented :

$$w = \begin{cases} w^- & if \quad \sigma < 0 \\ w^+ & if \quad \sigma > 0 \\ 0 & if \quad \sigma = 0 \end{cases} \tag{3.22}$$

with the trivial switching function

$$\sigma(v) = v - \tilde{v} \tag{3.23}$$

and where $\tilde{v}$ represents both the upper ($\overline{v}$) and lower ($\underline{v}$) limit of $v$. From Eq. 3.23 either the upper $\overline{\sigma}$ or lower $\underline{\sigma}$ switching function may result (see Fig. 3.9).

The first order filter is used to fulfill the unitary relative degree SM necessary condition and to smooth out the conditioning signal $w$ :

$$F(s) : \begin{cases} \dot{x}_f = \lambda_f x_f + w + r \\ r_f = -\lambda_f x_f \end{cases} \tag{3.24}$$

with $r$ being the original reference of the control system. The filter bandwidth should be higher than the one of the constrained system for the system response not to be unnecessarily degraded during unconstrained operation.

Finally, it is worth remarking that if the constrained system $S'_c$ was not biproper, the SMRC could always be applied by considering additional system states $\mathbf{x_s}$ in $\sigma$, so that the transversality condition (with respect to $w$) holds. This case is illustrated in Fig. 3.9 with the arrow labeled $x_s$ pointing to the block $\sigma$. For further details see [45].

## 3.4 SMRC application to dynamic AUV control

In this section a control algorithm, inspired on SMRC ideas, is developed to auto-regulate the AUV Ciscrea speed reference control in path following task, taking account the underlying constraints of its thrusters. Additionally, the corresponding simulations and its experimental validation is presented.

### 3.4.1 AUV Ciscrea path motion conditioning

Here, the SMRC is not used exactly as presented in the theorical approach. Indeed it has been modified in order to better fit with the path following issue. The proposal consists in adapting the speed of the path reference for the AUV Ciscrea when a given controller reaches its propeller actuator limits. For this, it is started from the premise that the path is parameterizable and continuous, it means that it is possible to express the path reference $\eta_{ref}$ and its first derivate as :

$$\eta_{ref} = f(\lambda) \qquad \dot{\eta}_{ref} = \frac{\partial f}{\partial \lambda}\dot{\lambda} \tag{3.25}$$

where $\lambda$ is the parameterization, and $\dot{\lambda}$ can be considered as the speed of the path reference. Taking this into account, it is possible to modify the scheme of SMRC as it is shown in Fig. 3.10. In this case, for the sake of clarity, only the heave direction is considered from the model proposed in section 3.2.

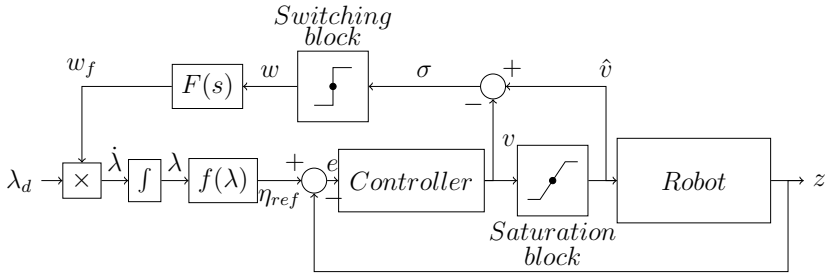



FIGURE 3.10 – Path motion conditioning for constraint mitigation scheme

Focusing our attention on the Fig. 3.10, it is possible to see a main block called “Robot" which represents the robot model in the heave direction, where its input is the digital torque command for the vertical thrusters $v$ and the output $z$ is the depth of the AUV. Naturally the input to the system is limited to its maximal values (due to

physical or energetic reasons), which is accounted for with the saturation block. This constraint is exploited to generate the $v$ variable and its restricted version $\hat{v}$. As our aim is mitigating the actuator saturation effects, these variables define the SM switching function as $\sigma = \hat{v} - v$. The surface associated with this switching function will generate a discontinuous signal $w$ :

$$w = \begin{cases} 1 & if \quad \sigma = 0 \\ 0 & if \quad \sigma \neq 0 \end{cases} \tag{3.26}$$

Differing from the traditional SMRC, $w$ is smoothed through a low pass filter "$F(s)$" to modify the speed reference parameter $\lambda_d$ instead of the reference itself, resulting in a motion parameter $\dot{\lambda}$ :

$$\dot{\lambda} = \lambda_d w_f \tag{3.27}$$

Once the reference $\eta_{ref}$ is generated through the integration block and the path generation block "$f(\lambda)$", it is compared with the actual robot position $z$ in order to generate the error signal for the controller block. The latter will generate the control signal for the robot.

In few words, the operation could be summarized as follows : when actuators are in their linear region, the SMRC auxiliary loop stays inactive and the reference speed is $\lambda_d$. When saturation limits are reached, $w$ switches between one and zero as fast as necessary to decrease the reference speed and avoid the controller surpassing those limits. Finally, if the condition is over, the SMRC loop turns into the inactive condition.

The following parameters should be considered for the control tuning :

— $\lambda_d$ : this parameter is the reference speed during the inactive condition of the auxiliary loop. Differing from more conservative strategies, here it must be chosen large enough to force the saturation of actuators, at least once in the path to be followed.

— $F(s)$ : the filter cut-off frequency should be taken high enough to allow fast stops of the reference, but sufficiently low to smooth the discontinuous signal $w$, in order not to produce chattering effect on the path reference [45].

— $\sigma$ signal must have relative degree equal to one with respect to the discontinuous signal $w$. This is a necessary condition for the SM establishment. In this case as long as it is used a controller with a derivative component this condition is guaranteed (if this were not the case, extra states should be considered in the

switching function). Considering a classical PD controller of the form :

$$v = K_p e + K_d \dot{e} \tag{3.28}$$

it is possible to get an expression of $\dot{\sigma}$ as :

$$\dot{\sigma} = \beta(\dot{e}, \ddot{z}, \lambda_f) - K_d \lambda_d \lambda_f \frac{\partial f(\lambda)}{\partial \lambda} w \tag{3.29}$$

On the right side of this expression two terms appear, one depending on $w$ guaranteeing the necessary condition provided $K_d$ is different from zero, and the other term $\beta$ is a function of the error derivate $\dot{e}$, the robot acceleration $\ddot{z}$, and the bandwidth of the low pass filter $\lambda_f$. As SM can be established provided the discontinuous term containing $w$ can change the sign of $\dot{\sigma}$ (recall Eq. 3.15), this means that both $\dot{e}$ and $\ddot{z}$ must be bounded, which is always true in practice. In chapter 6, numerical techniques will be addressed to obtain regions where the fulfillment of these conditions is guaranteed.

It is worth mentioning that in this approach, differing from the traditional SM developments, the switching signals are restricted to the digital implementation of an auxiliary loop. As a consequence, the proposal can be added to any pre-existing controller, and the commanded signal to the actuators is not a switching one, thus avoiding one of the main drawbacks of traditional SM designs : the chattering phenomenon.

### 3.4.2 Simulations

The simulation objective is to compare the performance achieved by the proposed methodology with a classical pre-designed PD action considered as a baseline controller. The simulations are performed for the Ciscrea heave direction only but they can be performed in the other axis. The simulation with the PD controller with a constant speed for the path reference is compared to the simulation achieved with the same controller when the quasi-sliding mode variable speed technique explained in the previous section is added.

It is important to emphasize for the clarity of the explanation, that a sinusoidal path in the heave direction is chosen (Eq. 3.30), thus this reference do not have a velocity

profile to be fulfilled as in the case of a trajectory (refer to subsection 2.4.1).

$$z_{ref} = -0.65\cos(\lambda) + 1.3 \tag{3.30}$$

Assuming that tuning of the embedded PD controller of the robot cannot be modified, the reference speed $\dot{\lambda}$ is changed as a tuning parameter. This value is chosen such as the actuators are on the border of saturation, as can be seen in Fig. 3.11. In this figure, the digital torque command is plotted for both cases the traditional PD approach (pd) and the proposal (sm), together with the torque constraints.

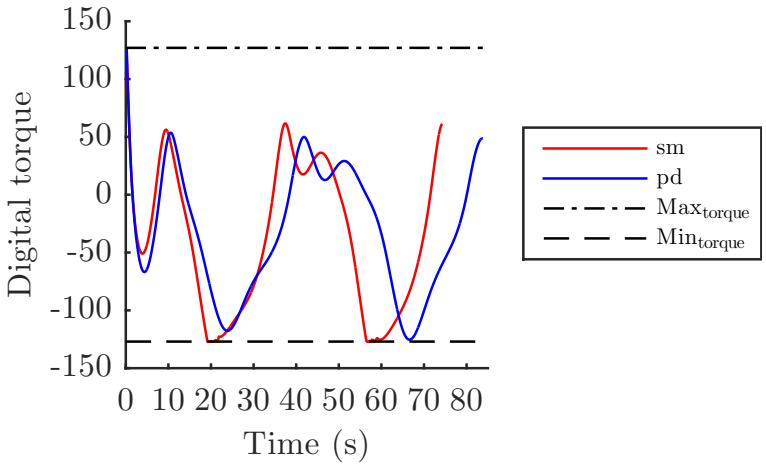



FIGURE 3.11 – Torque comparison simulation.

In contrast, for comparative purposes during the SMRC tuning, $\lambda_d$ is chosen to have a similar bounded position error as for the classic PD controller implementation, see Fig. 3.12.

A set of simulations were run and the results are given in Fig. 3.11 to 3.14. The following paragraphs describe the outcomes and show how the path motion conditioning approach improves the performance of the control.

Fig. 3.13 shows the depth time evolution with ($z_{sm}$) and without ($z_{pd}$) the SMRC, together with their corresponding path references ($zref_{sm}$ and $zref_{pd}$, respectively). It is possible to appreciate that the time evolution of the reference is not the same for both techniques, but spatially it is the same. Actually, we have a fixed speed reference for the PD control, and a reference with variable speed due to the SMRC loop. The path motion conditioning allows accelerating the reference as long as no saturation over the actuator exists, and when saturation arrives, it is slowed down (see times 19 s

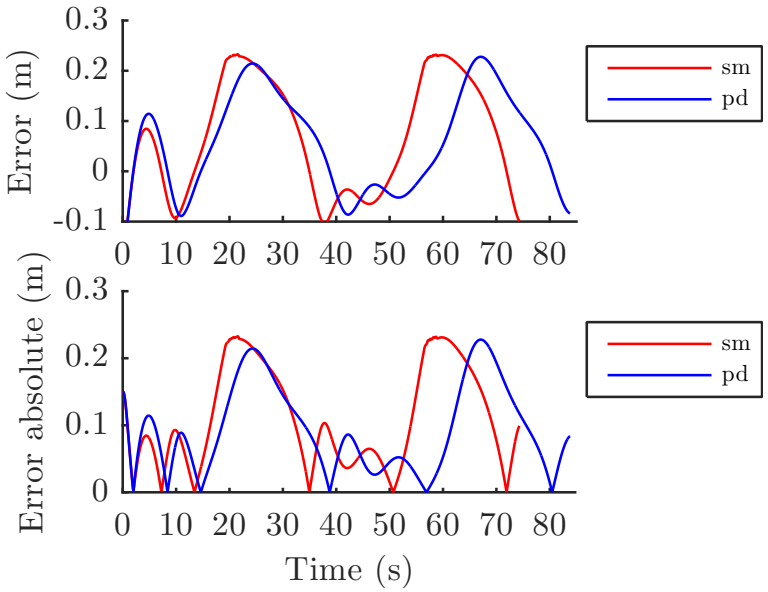



FIGURE 3.12 – Error comparison simulation.

to 22 s and 56 s to 59 s). In this way the proposal exploits better the operating range of the actuators. Furthermore, for the same error tolerances the constraint mitigation algorithm allows completing the path 10.5 seconds faster, which represents a 12.8% improvement in time. Similarly, it could improve path error if the same path following time were set for both cases.

Fig. 3.14, displays the remaining signals of the SMRC loop. Between 0 seconds and 19 seconds, no saturation phenomenon occurs, so the SMRC mitigation algorithm is turned-off and the path speed is fixed at $\lambda_d$. From time 19 s to 22 s the robot enters a closer path section where the speed imposed by $\lambda_d$ can not be followed. Then, the SMRC makes $w$ switch to slow down the reference so that the controller does not exceed the saturation limits. This can be verified in Fig. 3.13 where a *bump* in the SMRC reference can be seen. Also, note in Fig. 3.11 that the path reference speed generated by the SMRC loop is the maximum one that avoids exceeding the saturation limits. Finally, the SMRC becomes inactive again until time 56 s, when a similar speed adaption happens. The tuning of parameters involved in these simulations are listed in Table 3.5.

From the results of simulations, an improvement in travel time with SMRC is noticed. This result may mask the main advantage of the proposed method. For this, the Table 3.6 has been drawn up to highlight the benefits of the method. This table com-

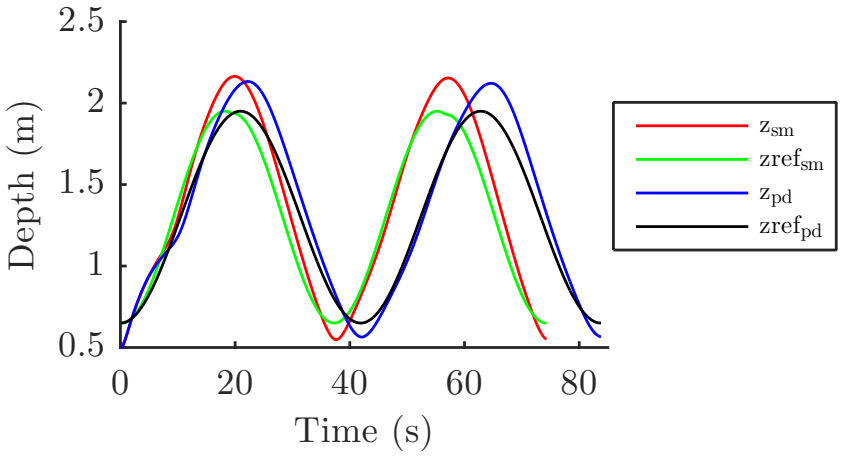



FIGURE 3.13 – Depth reference signal and path followed output for PD controller with and without path motion conditioning.

TABLE 3.5 – Simulation parameters corresponding to Fig. 3.11 to 3.14

| Parameter | Value |
|---|---|
| Speed of reference PD technique | $\dot{\lambda} = \lambda_d = 0.15$ |
| Maximum speed reference SMRC technique | $\lambda_d = 0.175$ |
| Cutoff frequency of the low pass filter | $f_c = 2$ Hz |
| Sample time | $\Delta_t = 0.1$ s |
| Controller | $K_p = 541.43$, $K_d = 250$ |

pares travel times and errors involved in several simulations. The first two columns (PD&SMRC, PD) show the results of the previously explained simulations. Again, from Fig. 3.13 it is noticed that the two reference “in time" are not the same, as the PD reference speed is constant and as fast as possible to avoid open-loop operation due to actuator saturation. This could conduct to think what would happen if it is choosen a reference speed which speeds up the fixed controller PD. This is done in the simulation which results on the data of the third column in the table (PD $\lambda_{dt}$), where it is possible to appreciate that in these conditions a greater error is found apart from open-loop operation (time during which the controller output exceeds the actuator limit).

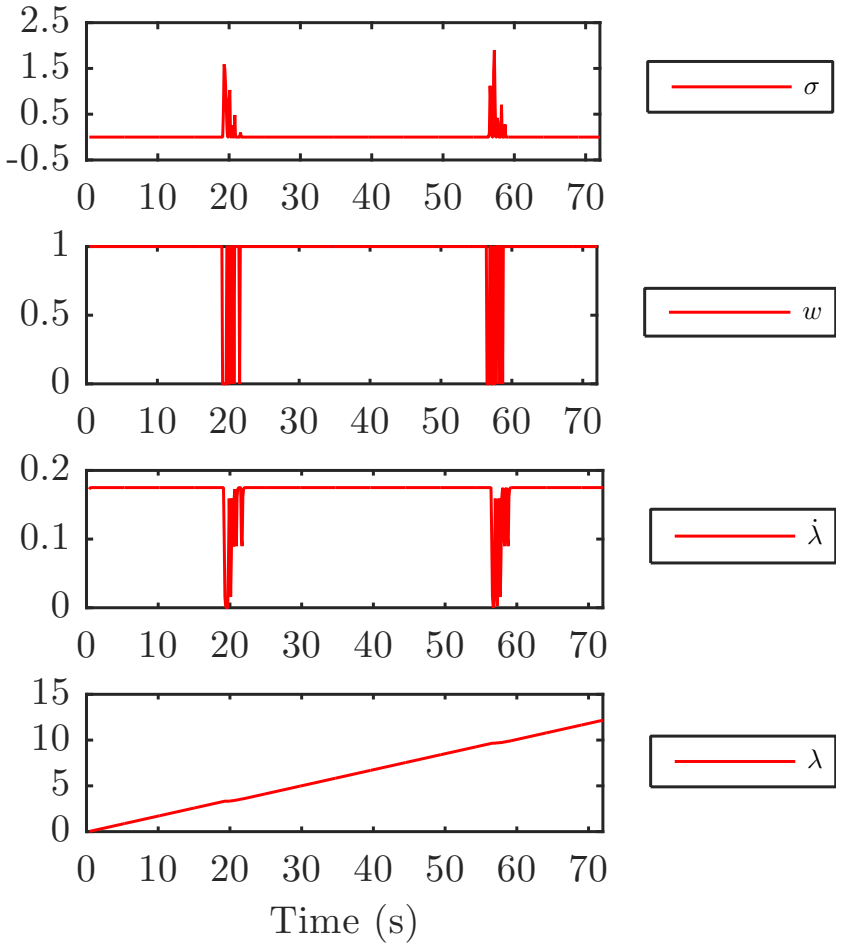



FIGURE 3.14 – SM and auxiliary signals simulation.

TABLE 3.6 – Comparative simulations

| | PD&SMRC | PD | PD $\lambda_{dt}$ | PD+ $\lambda_{dt}$ |
|---|---|---|---|---|
| Travel time [s] | 74.3 | 84.8 | 74.3 | 74.3 |
| Maximum absolute error [m] | 0.23 | 0.23 | 0.30 | 0.24 |
| RMSE [m] | 0.12 | 0.11 | 0.15 | 0.12 |
| Percentage of time in open-loop due to saturation | 0% | 0% | 11.64% | 9.65% |

The next step in this logic of reasoning would be to increase the aggressiveness of our main controller. The simulation for a 20% of increment in the PD coefficients results in the data showed in the PD+ $\lambda_{dt}$ column. We can notice the same error as for the original PD with the SMRC, but with a relatively long open-loop operation due to actuators saturation. In the last simulation, to obtain the same degree of error as the proposed technique, it was required to re-tune the main controller and to a-priori adjust the reference speed. This is an extra advantage of the SMRC, by adapting the speed of the reference for a given controller the performance improvement of potential re-tuning is automatically achieved.

### 3.4.3 Experiments

To complete the analysis of the proposed control algorithm, experiments in a pool have been performed and this section presents the results of the experimental test. For this case similar results are obtained with slight differences due to the real dynamics of the robot. The tuning of the parameters involved are listed in Table 3.7.

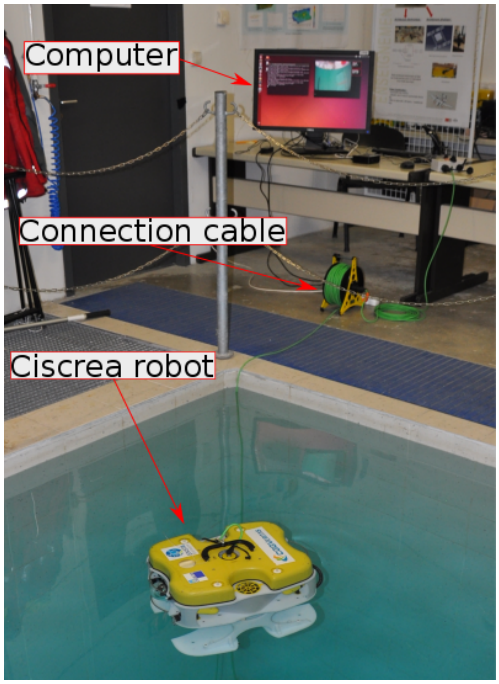



FIGURE 3.15 – Ciscrea setup at ENSTA Bretagne.

It is meaningful to note the difference in the cutoff frequency and $\lambda_d$ parameter with

respect to the simulation case. This is mainly to compensate for uncertainty and noise present in the real robot, which may affect the SM existence condition (see Eq. 3.29). The experimental setup at ENSTA Bretagne used to perform these experiments is shown in Fig. 3.15.

Figures 3.16, 3.17, 3.18, and 3.19 show similar results to the previous section (the same colour lines are used). The main difference that can be noticed is that in this case the sliding regime is longer than in the simulation, as it is now established between times 20 s to 35 s and 58 s to 72 s. This is attributable in part to the lower cutoff frequency in the low-pass filter and the high frequency components neglected in the robot modeling.

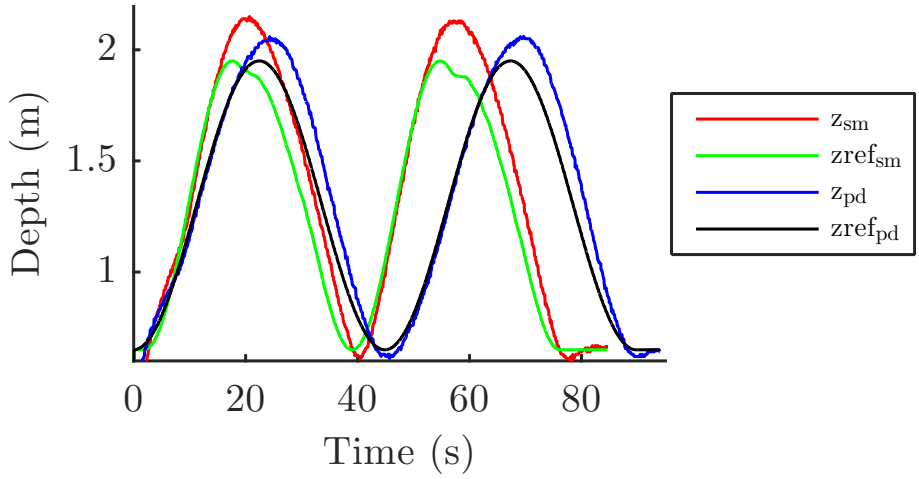



FIGURE 3.16 – Experimental depth evolution with (red) and without (blue) SMRC constraint mitigation, and their corresponding path references.

The experimental results show that the SMRC technique fulfills the torque constraints (Fig. 3.19) mitigating their effects on closed-loop performance. Indeed, given the same error tolerance (Fig. 3.17) the path is effectively completed in a shorter time when the SMRC path reference adaption is added to the original controller (Fig. 3.16).

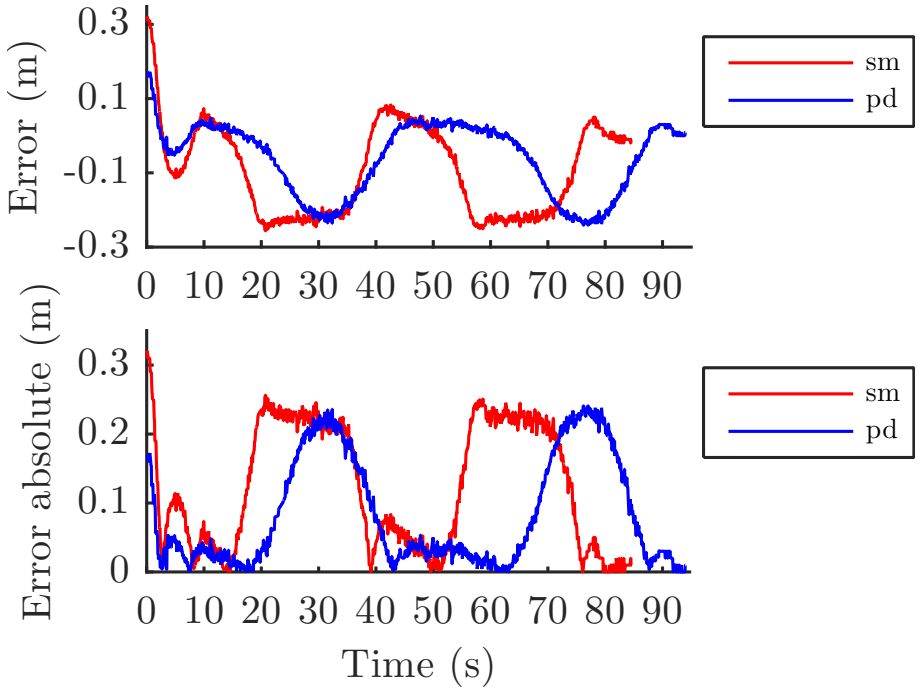



FIGURE 3.17 – Experimental path error with (red) and without (blue) SMRC constraint mitigation

TABLE 3.7 – Experimental parameters

| Parameter | Value |
|---|---|
| Speed of reference for PD technique | $\dot{\lambda} = \lambda_d = 0.15$ |
| Maximum speed reference for SMRC technique | $\lambda_d = 0.2$ |
| low pass filter (Cutoff frequency) | $f_c = 0.24$ Hz |
| Sample time | $\Delta_t = 0.1$ s |
| Controller | $K_p = 541.43$ and $K_d = 250$ |

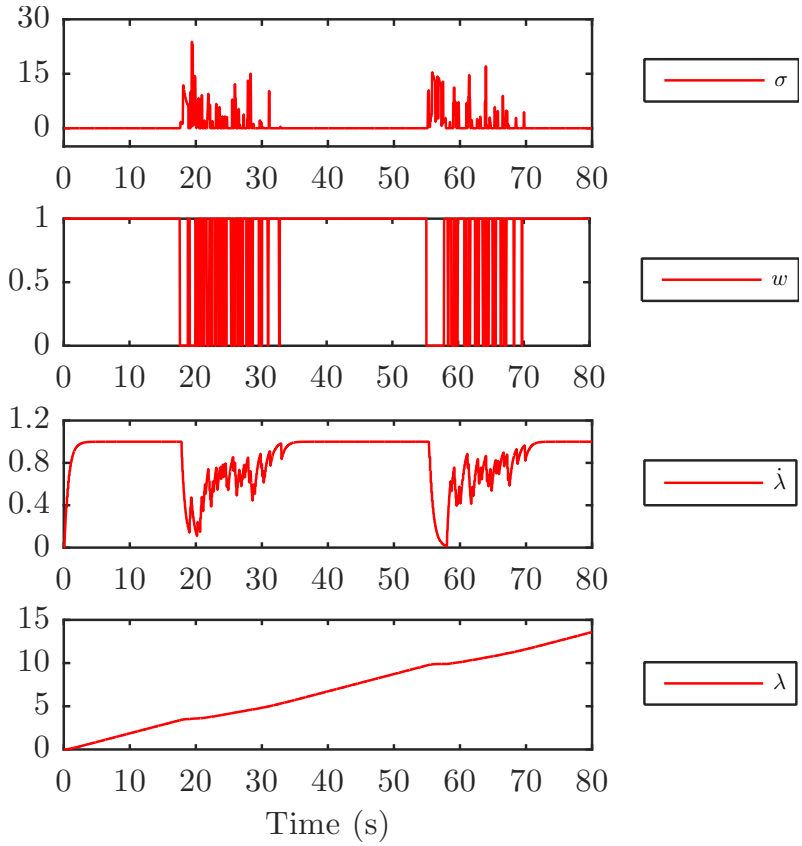



FIGURE 3.18 – Experimental SMRC signals

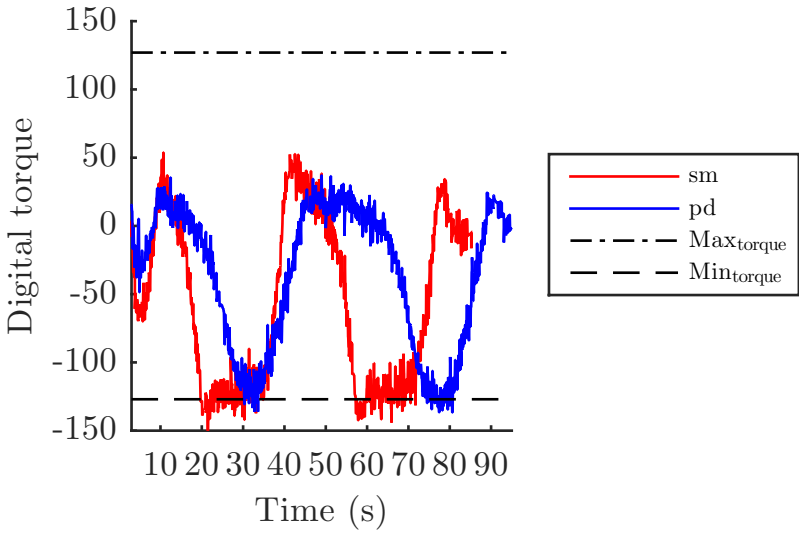



FIGURE 3.19 – Experimental torque comparison

## 3.5 Conclusions

In this chapter, a proposal has been presented to mitigate the problem of input constraints in path following applications. The proposed technique is simple to be implemented in real time applications and at the same time it presents robustness features inherited from its SM operation.

When the system falls in an input saturation, the proposed technique through the implementation of an auxiliary loop, forces the system to stay at the limit of the saturation region. To do this, the technique modifies the reference, assumed generated by a higher level, in order to make it compatible with the input constraints.

As case study, input restrictions over an AUV in path following application has been analyzed. From both simulation and experimental validation it is possible to conclude that the path motion conditioning applied to the dynamic control of the AUV effectively mitigates the actuator saturation effect on path following task, achieving a performance improvement in the time response.

Although the chapter has focused on a practical application, it should be noted that the methodology presented can be extended to any system with input constraints, which may be physical or virtual (for example constraints due to energetic criteria).

The work developed in this chapter has led to the presentation of three conference papers [46] (with its publication in IFAC-Papers-OnLine), [47], [48] and a journal publication [49].

CHAPITRE 4

# OUTPUT-CONSTRAINED ROBOTIC CONTROL

In the previous chapter the problem of input constraints in robotic systems was addressed. Here, in contrast, we will deal with constraints on the system outputs and their effects on the overall performance, recall the problems on robotic control described in section 2.5. Limitation in the workspace for robot manipulators, dynamic obstacles for mobile robots, or a maximum dynamics for transport situations are examples of constraints on the system outputs.

How to deal with these constraints depends in part on the application. Some previous approaches have been the use of predictive techniques, control through centralized systems in the case of robotic swarms, restricting inputs from the system itself, etc. Here it is proposed an external loop that, when detecting the limit condition, forces the behavior of the system to fulfill the given constraint. This chapter focuses on strict path tracking applications, which are a particular case of this type of constrained system.

## 4.1 Case of interest : strict path following

Apart from path following (seen in chapter 3), path planing and obstacle avoidance are also among the most frequent tasks commissioned to robots. Usually these duties are studied in an isolated way passing over their strong coupling. In general, during a path following task in an unstructured dynamic environment, when a collision situation is detected, a path re-planing must be done reconsidering the new state condition. This re-planing must be done merging global information and local information, as has been presented in subsection 2.4.2. In part, to overcome the collision situations several obstacle avoidance methods exist, with the most extended ones being :

1. The Potential Field Methods (PFM) : the robot is treated like a particle under the influence of an artificial force field where the obstacles exert repulsive forces, while the target applies an attractive, the sum of all forces determines the direction and speed of travel. This is the most extended method due to its easy on-line implementation. However, some drawbacks of this method are trap situations due to local minima, no passage between closely spaced obstacles, and oscillations [23].
2. Vector Field Histogram (VFH) : the method uses a polar histogram constructed around the robot, where each component represents the obstacle polar density in the corresponding sector. The set of candidate directions is formed with the components of lower density than a given threshold, and closest to the component that contains the target direction. Finally, through heuristics the robot direction is selected. The VHF is a method formulated to work with probability obstacle distributions and thus is well adapted to work with uncertain sensors such as ultrasonic sonars. One of the drawbacks is the computational cost of the method, although some simplifications have been proposed [24].
3. Velocity obstacles (VO) : this method forms a set of candidate control signals that are within the maximum speed of the vehicle. These signals generate safe trajectories considering the obstacle speeds and can be reached in a short period of time given the vehicle acceleration. From this set, one control signal is selected from the maximization of an objective function. The main advantage of this method is that it takes into account the obstacle velocities, thus it is well suited to dynamic scenarios [25].

As it has been remarked previously, these methods are the most widespread ones to avoid obstacles, but they share the disadvantage that they need to abandon the pre-established path to avoid collisions. There are multiple applications in which this disadvantage makes its application impossible. Such situations of strict path following are those which the technique proposed in this chapter seeks to solve. Although this seems a very strong constraint, it is a situation found in several applications like industrial line following robots or in automated warehouses [7], and it is not exclusive for robotics : other fields share the interest in this problem as the optimization of railways operations [50] or the recently presented world's first virtual track train. The strict path following has been less studied in literature than the general case where the path is not strict. Usually, the way this topic is addressed is through the analysis of collision situa-

tions in multi-robots operation at constant speeds when points of the path are common to more than one vehicle [51].

Here, it is proposed a new computationally non-expensive method for path following (Collision Avoidance Speed Adaption, CASA), that imposes a desired dynamics over the vehicle when a collision situation arrives. The main idea is to extend the path motion parameter adaption proposed in the previous chapter to deal with output constraints. For illustrative purposes, the method is evaluated in simulation first with a differential mobile robot and then using the experimental AUV Ciscrea robot. In the next section a definition of the obstacle avoidance problem is done, then in section 4.3 an explanation of the proposal is presented, while sections 4.4 and 4.5 illustrate the application of the proposal through simulations.

## 4.2 Problem description

It is defined the general problem of vehicle navigation through a given path with collision avoidance. It is assumed that there exists a dynamic environment that is unknown to the robot and where a pre-elaborated and parametrizable path is defined. This environment is conformed by moving or stationary objects, which are modeled as components of a time variant planar subset $\Psi$. For the practical point of view, it is supposed that the pre-elaborated path does not contain any collision situation with the stationary components of the environment, in other words the path is supposed realizable.

It is defined the distance $d(t)$ from the robot position $\mathbf{p}(t)$ to the environment $\Psi$ as :

$$d(t) := \min_{\mathbf{r}\in\Psi} \parallel \mathbf{r} - \mathbf{p}(t) \parallel \tag{4.1}$$

where $\parallel \cdot \parallel$ denotes the standard Euclidean vector norm, and $\mathbf{r}$ is the nearest obstacle's position belonging to the subset $\Psi$.

As it is desired to follow a path, it is defined a time variant target $\eta$ which is going to move through the path. The objective is that the robot follows this target through the path in a safe way, maintaining a $d_{safe} > 0$ distance to the obstacles in the environment, see Fig. 4.1. It is emphasized that the robot must not leave the path as a requirement of the application.

Furthermore it is demanded to the robot to follow a prescribed dynamics when it is approaching an obstacle, and to stop (or even to go back on the road) in case the $d_{safe}$

constraint cannot be accomplished.

In the next section the proposed method to fulfill the problem is presented.

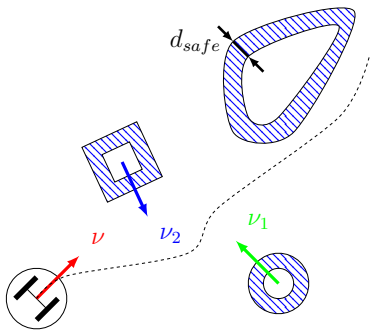



FIGURE 4.1 – The environment with the obstacle subset $\Psi$ and the enlarged subset $\hat{\Psi}$ with the $d_{safe}$ neighborhood

## 4.3 Collision Avoidance Speed Adaption (CASA)

The presented problematic is tackled here with an auxiliary loop that modifies the reference speed of the robot when potential collisions are detected. The scheme in Fig. 4.2 shows the block diagram of the proposal, which is called Collision Avoidance Speed Adaption (CASA). It is assumed that the robot control, as could be a traditional PID control, is implemented inside the block called “Robot + Robot Control”. Also it is considered that a first derivative feedforward action takes place inside it, as it is commonly used for robotic reference tracking [52]. In addition, it is supposed that the path is previously generated by a superior control level, and it is parameterizable. The parameter $\lambda$ commands the speed of advance from the path generator represented by the block “$f(\lambda)$”. It is worth mentioning, that the proposed approach can also be applied to those problems in which the path is generated “on-line" as the robot moves, e.g. in line followers robots.

The key of CASA method is to design a sliding surface associated to a discontinuous law (Eq. 4.3), which generates the motion parameter over the path. This ensemble is going to define the dynamics followed during the collision situation. It is understood as a collision situation when the distance and the approaching speed between the robot and a potential obstacle overpass a given maximum approaching dynamics.

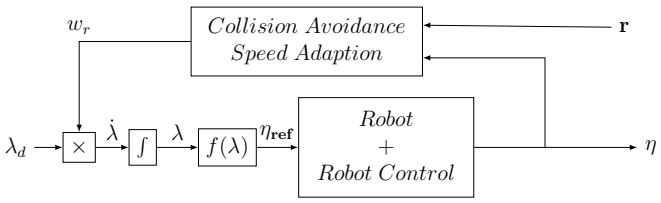



FIGURE 4.2 – Auxiliary loop proposed as an obstacle avoidance technique, based on SM

To this end the sliding mode (SM) surface $\sigma = 0$ is proposed with :

$$\sigma = d_{safe} - k_d d - k_{dd}\dot{d} \tag{4.2}$$

which depends on $d$, $\dot{d}$ and the weighting parameters $k_d$ and $k_{dd}$. The last ones define the desired approaching dynamics to the minimal distance $d_{safe}$ constraint. To complete this formulation the associated switching function is defined as :

$$w_r = \begin{cases} 1 & \sigma \leq 0 \\ b & \sigma > 0 \end{cases} \tag{4.3}$$

where $b$ can take the fixed values $0$ or $-1$. When parameter $b$ is set to $-1$, the robot is allowed to reverse its direction if necessary to avoid a collision, while when $b$ is set to 0, the robot is just allowed to stop over the path until the risk of a collision disappears.

From Eq. 4.2 and Eq. 4.3 it is possible to see that a discontinuous signal $w_r$ is generated. This signal can be smoothed through a first-order filter, if the particular application so requires it. If it is done, the signal $w_{rf}$, a soft version of $w_r$ is generated. This filter could be described as :

$$\begin{cases} \dot{x_f} = \lambda_f x_f + w_r \\ w_{rf} = -\lambda_f x_f \end{cases} \tag{4.4}$$

Note that this filter is necessary as long as the block “Robot + Robot Control” has input-to-output relative degree zero, for $\sigma$ to have relative degree one with respect to the discontinuous signal $w_r$.

Finally the speed of the commanded reference, with the filter interleaved is produced from this soft version of $w_r$, after being affected by $\lambda_d$ as :

$$\dot{\lambda} = \lambda_d w_{rf} \tag{4.5}$$

The tuning parameter $\lambda_d$ will determine the maximal speed reference of the path under normal conditions, it means without potential collisions. This new signal $\dot{\lambda}$, the adapted motion parameter, is integrated in order to generate $\lambda$ and feed the path generator block.

In the operation of the system there are two possible situations. In the first situation no collision is detected, so the robot follows a given pre-elaborated path while the speed adaption loop rests inactive, that is the $w_r$ signal is equal to "1". Here the dynamics of the system is governed by the main control of the robot. When a collision situation arrives, it means that the approaching dynamics is faster than the desired one ($\sigma = 0$, with $\sigma$ defined in Eq. 4.2), the discontinuous signal $w_r$ changes its value to "$b$" and then after passing through the low-pass filter affects the $\dot{\lambda}$ parameter slowing down the increase of the $\lambda$ parameter. Actually, during this condition a fast commutation of $w_r$ signal forces the system to follow the desired dynamics imposed by the sliding mode (SM) surface $\sigma = 0$. When the collision situation vanishes the system returns to the first situation. This fast commutation is possible just if $\dot{\sigma}$ depends on $w_r$, i.e. $\sigma$ has relative degree one with respect to $w_r$, which is a necessary condition for the SM establishment. In consequence, the system will slide over $\sigma = 0$ as long as the discontinuous signal $w_r$ is enough to change the sign value of $\dot{\sigma}$ from side to side of this surface.

The discontinuous signal $w_r$ slows down the reference speed in function of the approaching distance and its derivative. In the case this approach is sufficiently soft the vehicle will first break with the desired dynamics, then stop, and if the obstacle continues towards the vehicle, it will start going back according to the desired dynamics imposed by Eq. 4.2 (only for the case when $b = -1$). In extreme cases when the approach is too fast the speed adaption loop could not fulfill the desired dynamics (the SM could not be established). Instead, it makes the robot go back over the path when $b = -1$, or stops if $b = 0$.

Some extra considerations over the technique are :

— The choice of the cut-off frequency of the low pass filter ($\lambda_f$). On the one hand a too low bandwidth leads to slow reaction of the system in front of a sudden obstacle, and on the other hand a too high bandwidth will result in a non smooth rolling of the robot through the path. Finally, its optimal value depends on the expected speeds of the mobiles involved and the acquisition rate of the distance measures.

— The choice of $k_d$ and $k_{dd}$ is restricted to the desired dynamics. During the SM, the dynamic behaviour has an exponential form with a time constant $\tau = k_{dd}/k_d$, which must be realizable for the robot. Furthermore the $k_{dd}$ parameter must be different from zero in order to fulfill the necessary condition for the SM establishment.

— It is worthy to remark that all the high frequency switching in the proposal is restricted to a digital implementation of an auxiliary loop.

— The critical point in the application is the sampling time in the distance measurements. These measurements, through a $\dot{d}$ estimator, and the maximum dynamics of the robot will define the maximum speed of obstacles with which the robot can maintain the desired dynamics, and in the extreme case dodge them.

In the next section this technique is tested over different situations where it is possible to get a general idea of its behaviors, and the possible applications covered by it.

## 4.4 Application to a differential mobile robot

In this section several simulations are presented in order to show the potentiality of the proposal. The implementation of simulations has been made through Matlab environment and V-REP simulator [53]. The latter offers not only realistic simulation graphics but also the capability of considering the real dynamic properties of the robots. It is employed the Pioneer P3-DX robot which is available in the library of the program and a validated model.

FIGURE 4.3 – Pioneer P3-DX mobile robot

### 4.4.1 Robot and inner controller description

The Pioneer P3-DX mobile robot (Fig. 4.3) is a differential robot, which can be modeled as follows :

$$\begin{aligned} \dot{x} &= \nu \cos\theta \\ \dot{y} &= \nu \sin\theta \\ \dot{\theta} &= \omega \end{aligned} \tag{4.6}$$

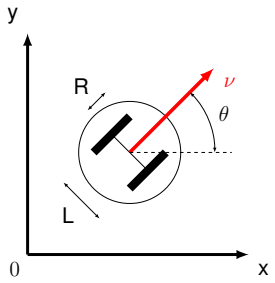



FIGURE 4.4 – Reference frame

Here $\mathbf{p}(t) = [x(t), y(t), \theta(t)]^\top$ is the vector of the vehicle's Cartesian coordinates and $\theta(t)$, its heading. The angle $\theta(t) \in (-\pi, \pi]$ is measured in the counter-clockwise direction from the x-axis, see Fig. 4.4. $\nu$ and $\omega$ are the linear and angular velocity respectively, both bounded variables.

For this model the control variables are $\omega$ and $\nu$, however in the real robot application is more often to command the angular velocities of its wheels $w_{right}$ and $w_{left}$. The relation between $\omega$ and $\nu$ to $w_{right}$ and $w_{left}$ could be written as in Eq. 4.7 and Eq. 4.8.

$$w_{right} = \frac{Lw + 2\nu}{2R^2} \tag{4.7}$$

$$w_{left} = \frac{2\nu - Lw}{2R^2} \tag{4.8}$$

Where R is the wheels' radius and L the distance between the two actuated wheels, see Fig. 4.4.

This model is restricted to the rolling without slipping kinematic constraint given by Ec. 4.9.

$$-\dot{x}\sin\theta + \dot{y}\cos\theta = 0 \tag{4.9}$$

The main control proposed is conformed by two independent proportional actions

for the $\nu$ and $\omega$ command signals, and a corresponding feedforward action. Fig. 4.5 shows the proposed configuration where $\eta = [x_r, y_r, \theta_r]^\top$ is the path reference, $\mathbf{e} = [x_r - x, x_y - y, \theta_r - \theta]^\top = [e_x, e_y, e_\theta]^\top$ is error vector, and $\mathbf{u} = \mathbf{u_c} + \mathbf{u_f} = [\nu, \omega]^\top$ is the control signal.

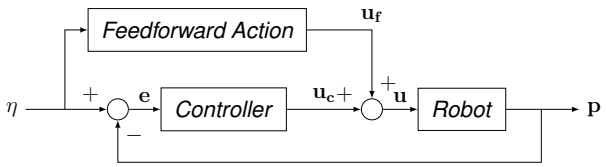



FIGURE 4.5 – Inner control for path following

The equation that governs the control is given by :

$$\mathbf{u_c} = \begin{bmatrix} k_{p\nu}\sqrt{e_x^2 + e_y^2} \\ k_{p\omega}\,\mathrm{atan2}(sin(e_a), cos(e_a)) \end{bmatrix} \qquad \text{with} \quad e_a = \mathrm{atan2}(e_x, e_y) - \theta \tag{4.10}$$

and the corresponding feedforward action by :

$$\mathbf{u_f} = \begin{bmatrix} k_{f\nu}\sqrt{\dot{e_x}^2 + \dot{e_y}^2} \\ k_{f\omega}\dot{\theta_r} \end{bmatrix} \tag{4.11}$$

$k_{p\nu}$, $k_{p\omega}$, $k_{f\nu}$ and $k_{f\omega}$ are the tuning parameters for the control set. Note that the function $\mathrm{atan2}(x, y)$ is the arctangent function where its result is the angle in radians between the positive x-axis of a plane and the point given by the coordinates (x, y) on it. No further details will be given about the controller, the interested reader can consult [4] where more details about the control and model can be found.

Notice also that from Eq. 4.2 to Eq. 4.11 it is possible to check the relative degree of $\sigma$ with respect to $w_r$, which is equal to one. Thus the necessary condition for SM is given.

### 4.4.2 Simulation results

In the following some simulation results are shown when parameter $b$ is set to $b = 0$, i.e. the robot is allowed to stop over the path when a collision situation happens, but cannot reverse its moving.

### Fixed obstacle

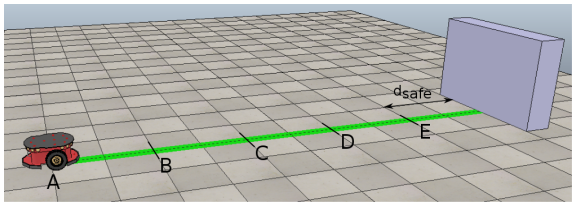



FIGURE 4.6 – Fixed obstacle situation

The first scenario shows the robot following a straight path which drives it against a collision with a fixed obstacle, see Fig. 4.6. At the beginning of the simulation our robot is too far from the obstacle so the speed adaptation loop rests inactive and the main controller governs the system. As the robot advances in the path the distance to the obstacle decreases. At the moment that the distance and its derivative break the maximum desired approaching dynamics, the SM is established. The auxiliary loop signals over time can be seen in Fig. 4.7.

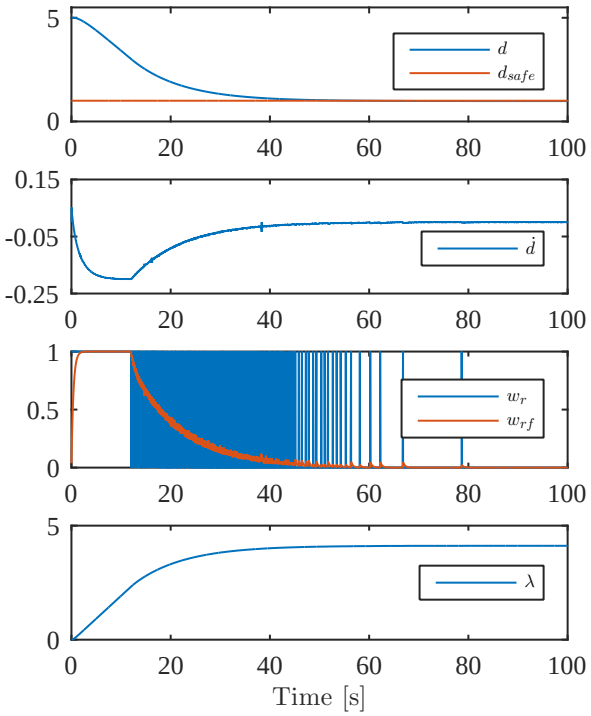



FIGURE 4.7 – Speed adaption loop's signals for a fixed obstacle

It is possible to observe at time 18 [s] that the SM starts, and consequently the

$\lambda$ parameter slows down its increase. Once the auxiliary loop is active, it forces the system to follow the desired dynamics slowing down the robot speed up to stop the robot just on the border of the safety region, where $d = d_{safe}$. For this simulation the tuning parameters were $d_{safe} = 1$, $k_d = 1$, $k_{dd} = 1$, sampling time $T_s = 10[ms]$, cut-off frequency of the low pass filter $fc = 0.4[Hz]$, and $\lambda_d = 0.2$.

It results interesting to observe the plane $d$ vs. $\dot{d}$ in Fig. 4.8, it shows how the system evolves for different path speed set-points. Lets pay attention to the "blue line". First, the robot starts from a rest condition at point "A" and evolves with the main controller dynamics. When the robot reaches the desired approaching dynamics, "C" point, the speed adaption loop is activated an forces the system to follow the sliding surface, which is represented by the line which crosses the "E-C" points. The system continues with the desired dynamics up to the point "E" which represents the limit of the allowed distance with the approaching speed equals to zero.

To show the strength of the proposal also in Fig. 4.8 it is possible to see other two simulations where the speed of approach is different, in one case slower than the previous one with $\lambda_d = 0.1$ and in the other case faster with $\lambda_d = 0.3$. Both cases show the same behavior. Note that the greater the speed the greater the distance to the obstacle at which the adaptive loop starts acting, see Fig. 4.6.

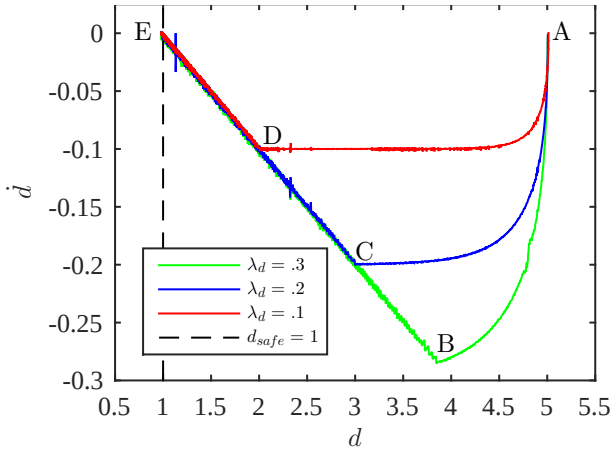



FIGURE 4.8 – Different speeds during the approach to obstacle

One of the distinctive characteristics of the proposal is the possibility of imposing different desired dynamics. Fig. 4.9 shows the same simulation but with different dynamics imposed through the change of the $k_{dd}$ parameter. In this figure it is also possible to observe a limit of the technique : in the case of the green curve the demanded ap-

proaching dynamics (dotted line) is too fast for the system so the auxiliary loop works as an emergency break, the stopping dynamics being the fastest which the robot can follow. Naturally, this situation can be avoided as long as $k_{dd}$ is properly chosen according to the robot dynamics features and the expected obstacle speed in the case of moving scenario.

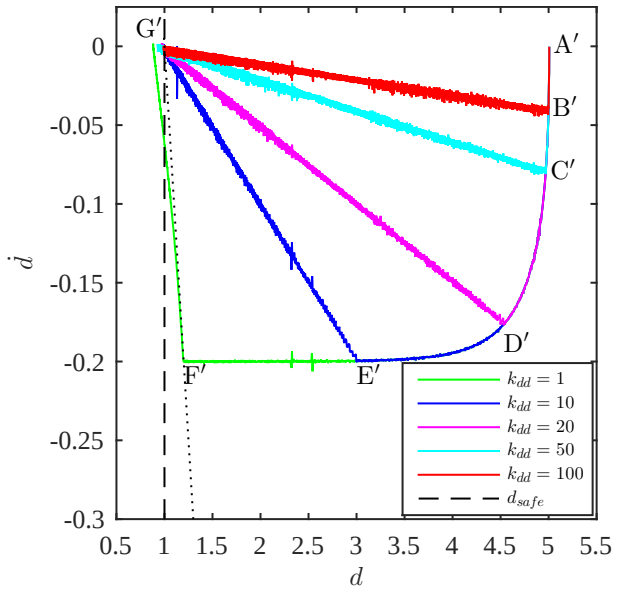



FIGURE 4.9 – Response of different dynamics imposed to the robot

### Moving obstacles

In this situation we are more concerned about the response of the system in the condition when moving obstacles are in the environment. In Fig. 4.10 the setup of the simulation is presented. Here we have a main robot with the proposed technique implemented which must follow a straight path (green one). Also we have two mobile obstacles which follow perpendicular paths to the first one. One of the obstacles is another Pioneer robot with a classical path following controller, which must follow the red path. The other obstacle is a human being who must follow the blue path, the last one moves faster than the robot.

Fig. 4.11 shows the speed adaptation loop signals. It is possible to see that the main robot starts its movement following the path according to the main controller dynamics. But soon at time 2 [s] the speed of approach and the distance to the first obstacle shoots the auxiliary loop slowing down the speed of the main robot, giving time to

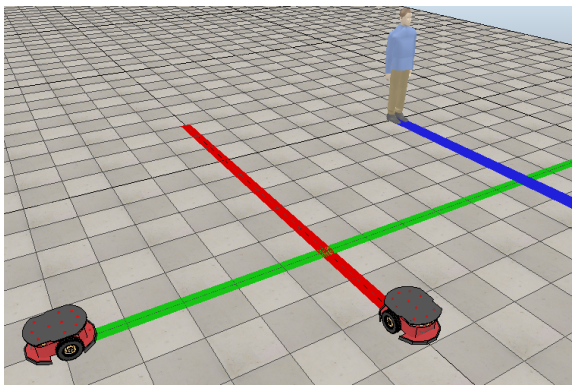


FIGURE 4.10 – Moving obstacles situation

the first obstacle to cross the path. Once the obstacle begins to increase its distance, the auxiliary loop, still active, allows the main robot to accelerate always following the desired dynamics up to the moment where the CASA algorithm loop changes again to the inactive state ($w_{rf} = 1$, $\dot{\lambda} = \lambda_d$).

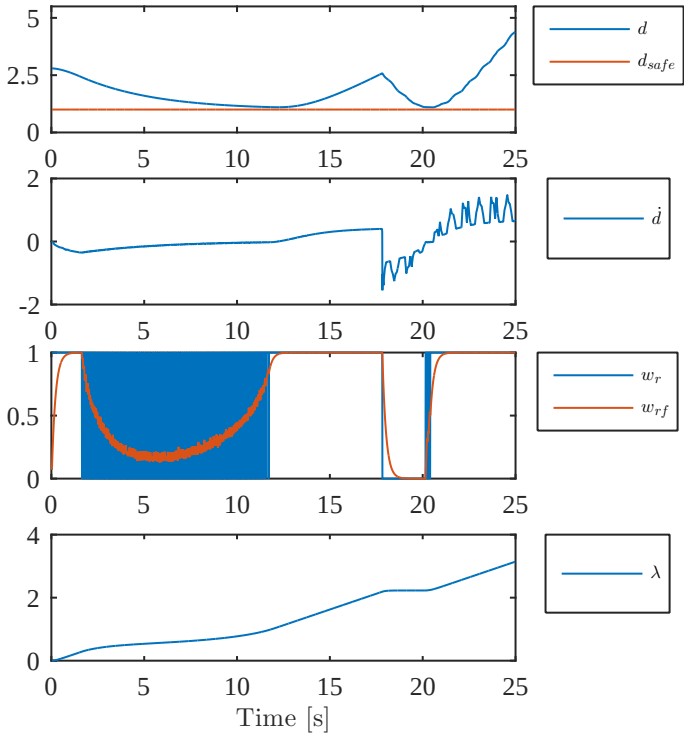



FIGURE 4.11 – Speed adaption loop's signals for mobile obstacles

Then the main robot faces another collision situation at time 18 [s], in this case is the human who approximates much faster than the previous obstacle so the speed adaption loop does not have time to establish an SM. In consequence, it acts as an emergency break which stops the robot allowing the human to cross the path up to the time 20 [s] when the human moves away from the robot position. In this last condition a SM is established when the robot starts to move again, and finally the inactive condition of the loop is reached. The main objective of this simulation is to observe the behavior in front of two obstacles moving at different speeds.

### Corridor situation

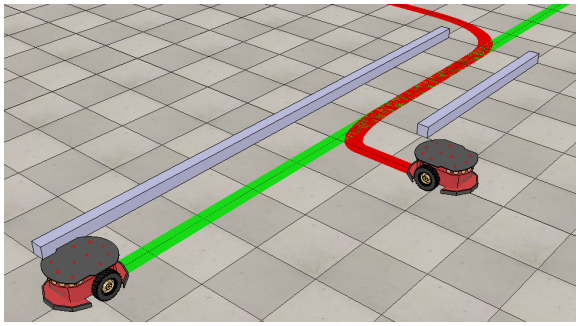


FIGURE 4.12 – Corridor situation

In this case the idea of the simulation is to test the ability of the proposed technique to adapt the robot speed to any other speed vehicle which must share the path followed. The setup of the simulation can be appreciated in Fig. 4.12. Here we have two vehicles, the main robot which have the proposed technique implemented and must follow the green path, and another robot (from here called the obstacle) with a traditional path following controller which must follow the red one. Both robots share a section of the path, that could be a corridor in a real factory situation. In addition to the previous description, the main robot can move faster than the obstacle, but is the second one which first reaches the corridor.

In Fig. 4.13 the speed adaption loop signals can be observed. Here at the beginning the robot starts to reduce its speed due to the proximity to the obstacle (time 1[s]). As soon as the obstacle gets into the corridor (time 8[s]), it increases the distance to the robot, so the auxiliary loop allows increasing the reference speed. The robot and the obstacle speeds can be seen in Fig. 4.13 as $\| \mathbf{v_r} \|$ and $\| \mathbf{v_{ro}} \|$ respectively. Once

both vehicles are in the corridor it is possible to see how the auxiliary loop modifies the speed of the robot to follow the obstacle just on the border condition of $d_{safe}$. As the obstacle must do a 90 degrees turn at the end of the corridor it must reduce its speed, and the speed adaption loop breaks the main robot (time 27 [s]), then it starts again the movement (time 30 [s]) with the prescribed SM dynamics until the CASA algorithm becomes inactive (time 34 [s]).

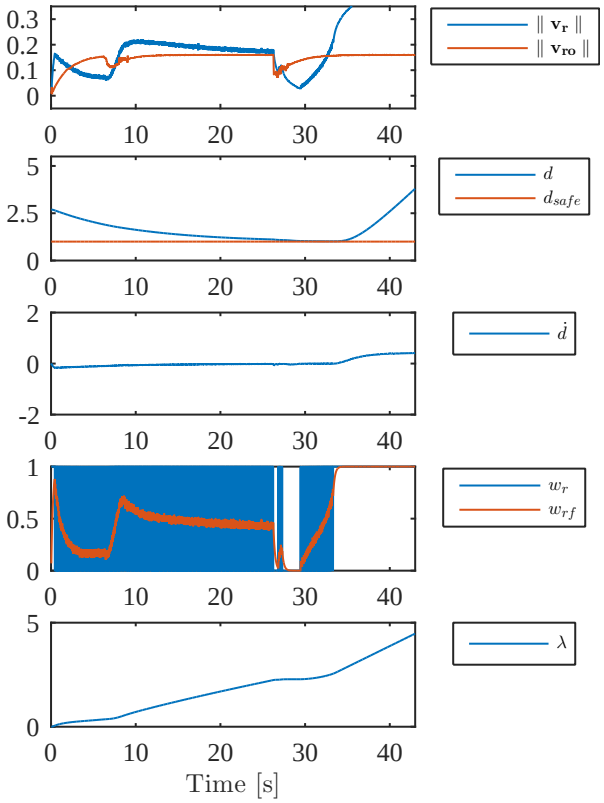



FIGURE 4.13 – Speed adaption loop's signals for a mobile obstacle in a hall condition

## 4.5 Application to AUV Ciscrea

In this section the application of the CASA algorithm (Section 4.3) to the AUV described in section 3.2 is done. Differing from the previous section, here the reference path is composed of three coordinates (surge, sway and yaw ) and the robot can reverse its direction of travel, i.e. the parameter $b$ is equal to $-1$. Also, as in this case the application does not require smooth movements, the first order filter of Eq. 3.24 is omitted.

### 4.5.1 AUV and inner control description

The AUV Ciscrea is used in these simulations, so the model described in section 3.2 apply here.

To implement a closed-loop position/orientation control, a simple approach using four independent stabilizing PID controllers and their corresponding feedforward actions in the Surge, Sway, Heave, and Yaw directions are used (Roll and Pitch are not controllable due to the disposition of thrusters). This is possible because of the low coupling among the outputs of the system. A schematic representation of this implementation can be seen in the Fig. 4.14.

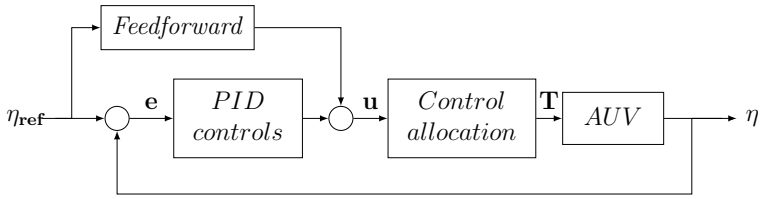



FIGURE 4.14 – Position/Orientation control of the AUV

The output of these actions ($\mathbf{u} = [u_x, u_y, u_{yaw}, u_{heave}]$) is affected by a control allocation stage (see Eq. 4.12) that maps these signals to the AUV thrusters command ($\mathbf{T} = [T_1, T_2, T_3, T_4, T_5, T_6]$) numbered as in Fig. 4.15.

$$\begin{aligned} T_1 &= -u_y + u_x - u_{yaw} \\ T_2 &= +u_y + u_x + u_{yaw} \\ T_3 &= -u_y - u_x + u_{yaw} \\ T_4 &= +u_y - u_x - u_{yaw} \\ T_5 &= T_6 = u_{heave} \end{aligned} \tag{4.12}$$

In the next subsection, this main controller is complemented with the proposed CASA algorithm in order to get an obstacle avoidance technique.

### 4.5.2 Simulations results

A test of the proposed algorithm is run here in Matlab environment. The objective of this simulation is to see the CASA algorithm adjusting the speed of movement of the AUV in a 3D application. In order to show these results, we pose a situation in which

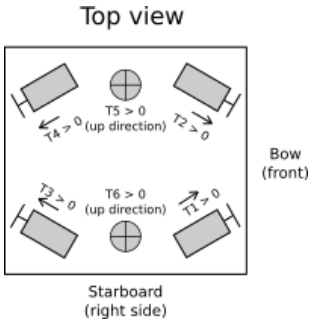


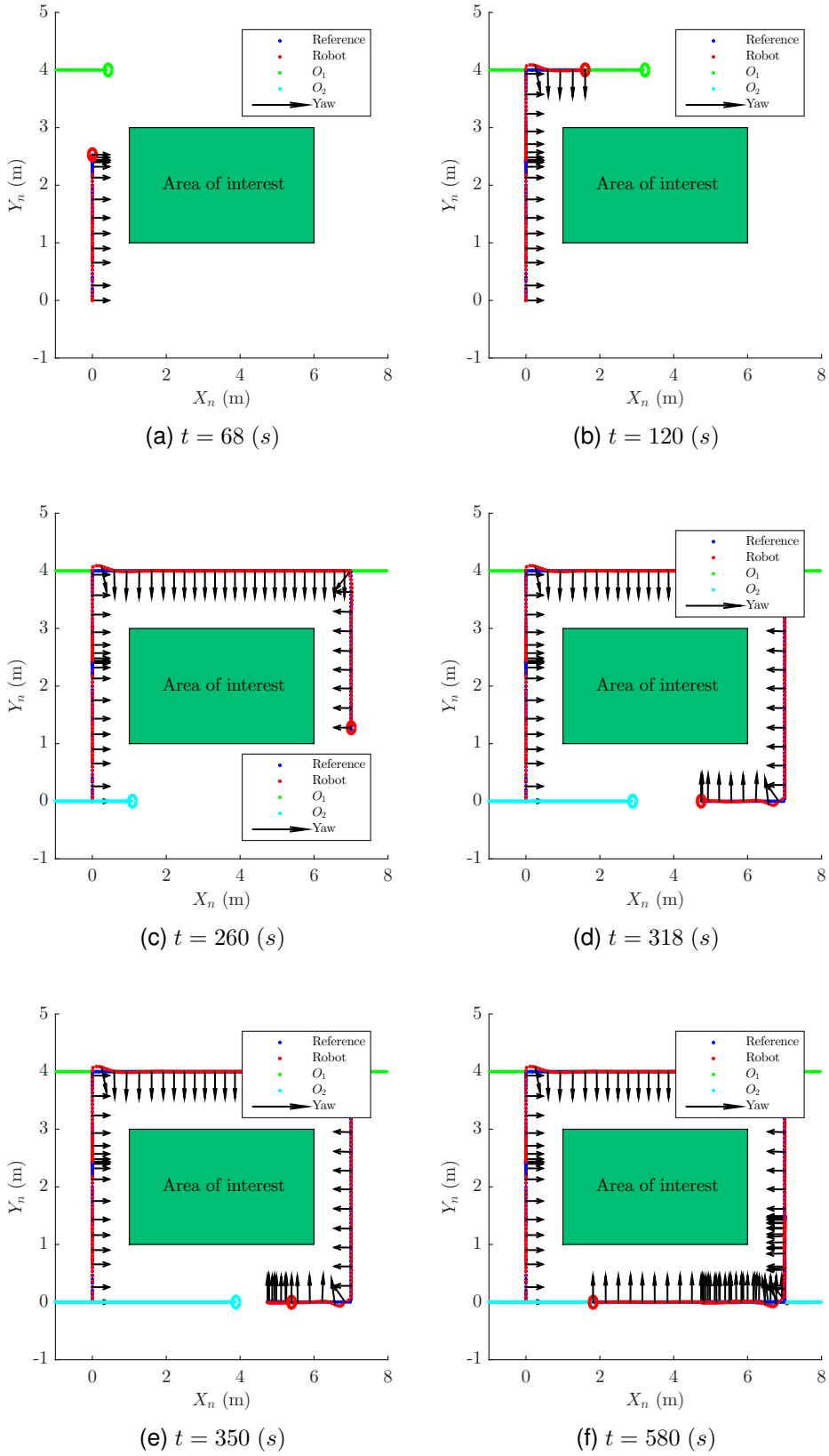

FIGURE 4.15 – Ciscrea's thrusters numeration

the AUV must travel the periphery of a rectangular area of interest, always with its bow pointing to its interior :

$$f(\lambda) = \begin{cases} x_{ref} = 0,\ y_{ref} = 0,\ \psi_{ref} = 0 \quad \text{if } \lambda \leq 0 \\ x_{ref} = 0,\ y_{ref} = 0.0667\lambda,\ \psi_{ref} = 0 \quad \text{if } 0 < \lambda \leq 60 \\ x_{ref} = 0.0667\lambda - 4,\ y_{ref} = 4,\ \psi_{ref} = -\pi/2 \quad \text{if } 60 < \lambda \leq 165 \\ x_{ref} = 7,\ y_{ref} = -0.0667\lambda + 15,\ \psi_{ref} = -\pi \quad \text{if } 165 < \lambda \leq 225 \\ x_{ref} = -0.0667\lambda + 22,\ y_{ref} = 0,\ \psi_{ref} = -(3/2)\pi \quad \text{if } 225 < \lambda \leq 330 \\ x_{ref} = 0,\ y_{ref} = 0,\ \psi_{ref} = -(3/2)\pi \quad \text{if } 330 \leq \lambda \end{cases}$$

In Fig. 4.16 a set of captures of the simulation is showed, where it can be appreciated that the workspace is shared by the Ciscrea AUV (represented by a red circle) and other two mobile objects ($O_1$,$O_2$) (represented by green and cyan circles respectively), which could be other AUVs working on the same area.

To facilitate the description of the simulation, it will be explained in five points indicated below. It is recommended that the reader see in parallel Figures 4.16 and 4.17 for a better understanding. In Fig. 4.17 it is possible to see all the internal signals involved in the CASA algorithm, while in each subfigure of Fig. 4.16 it is possible to see the area of interest, the components of the workspace, and the path followed by each robot. Also the bow direction of the AUV is indicated with a black arrow every 500 sample times over the path.




(a) $t = 68\ (s)$ (b) $t = 120\ (s)$

(c) $t = 260\ (s)$ (d) $t = 318\ (s)$

(e) $t = 350\ (s)$ (f) $t = 580\ (s)$

FIGURE 4.16 – AUV Obstacle avoidance

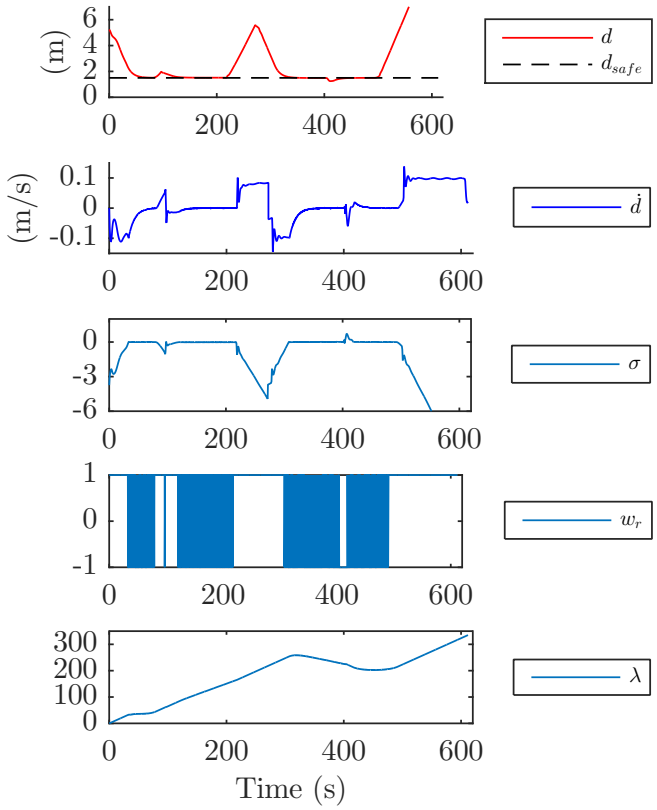



FIGURE 4.17 – CASA signals.

In the following a time sequence of the simulation is itemized :

— At the beginning, the AUV departs from the position (0,0), following a straight path in the $Y_n$-direction. As long as no collision situation arrives the speed of the reference, and therefore the AUV speed, is determined by the $\lambda_d$ parameter, as the signal $w_r$ is “1”. Around time 60 (s), see Fig. 4.16a, the AUV is approaching $O_1$, so the CASA algorithm is activated and a fast commutation is established in the signal $w_r$ (sliding mode over the surface $\sigma = 0$) slowing down the movement. Once this collision situation is released, the CASA algorithm returns to its inactive state $w_r = 1$ and the commanded speed is restablished.

— Some time later, in $t = 120$ (s), the AUV finds again $O_1$ but in this case both advance in the same direction ($O_1$ moves slower than the AUV, see Fig. 4.16b). The CASA algorithm modifies the speed of the AUV, in order to match the speed of both mobiles keeping the minimum distance $d_{safe}$.

— Once $O_1$ is far enough to disable the CASA algorithm, the robot will advance at the maximum speed fixed by $\lambda_d$, see Fig. 4.16c.
— After the last corner of the area of interest, the AUV advances through the path and it finds $O_2$ in the opposite direction, see Fig. 4.16d. In this case the CASA algorithm slows down the reference, up to the point where the condition $d \geq d_{safe}$ is violated. When this happens, the CASA algorithm reverses the reference over the path done, and consequently the AUV returns over this path, see Fig. 4.16e.
— Finally, after reversing the movement along the path, when the AUV finds its way free, the CASA algorithm is deactivated and the AUV runs over the path again with the speed fixed by $\lambda_d$, see Fig. 4.16f.

It results interesting to see the evolution of the robot's coordinates over time. Figures 4.18, 4.19, and 4.20 show the time evolution of the original reference without the CASA algorithm ($x_{ref}$,$y_{ref}$,$\psi_{ref}$), the reference affected by the CASA algorithm ($x_{SMref}$,$y_{SMref}$,$\psi_{SMref}$), the AUV position and orientation ($x$,$y$,$\psi$), and the obstacle positions $(x_{o1}, y_{o1})$ and $(x_{o2}, y_{o2})$. From these figures it is possible to see how the CASA algorithm modifies the path reference only "in time". It means that the original path generated by "$f(\lambda)$" is not modified in space but just in the time when the robot must go through it.

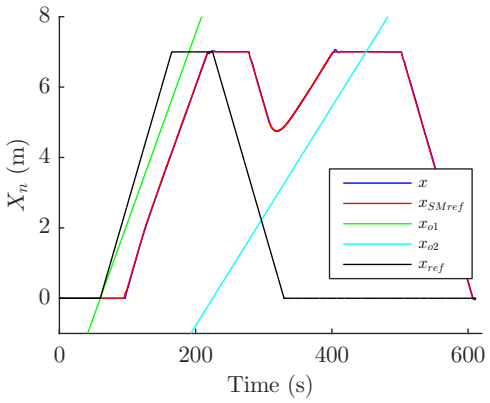



FIGURE 4.18 – Sway movement as a function of time.

To conclude the analysis of the proposed situation, in Fig. 4.21 the torque signal command of one of the horizontal thrusters is showed. This signal does not saturate the actuator (+/-127 value), but it is appreciable its high frequency component. These commutations of high frequency are due to the sliding mode operation of the CASA algorithm. In a real application, these commutations are going to be in the order of the

time step of the logical implementation usually much smaller than the actuator time response, so they will be filtered by the actuators.

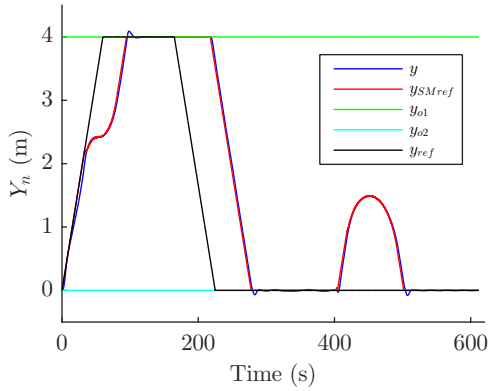



FIGURE 4.19 – Surge movement as a function of time.

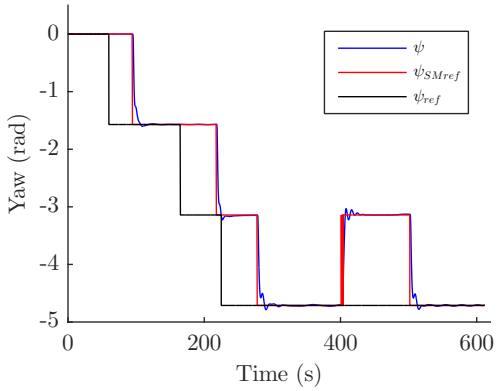



FIGURE 4.20 – Yaw movement as a function of time

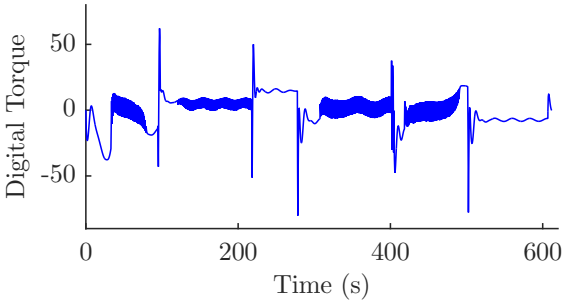



FIGURE 4.21 – Torque.

## 4.6 Conclusions

In this chapter, it has been developed a technique based on an auxiliary loop to deal with collision avoidance (output constraints) in robotic systems, under strict path following task. In particular, the technique modifies the reference of the system temporarily (not spatially) to achieve compliance with the restrictions imposed on its outputs.

Simulations of the two analyzed application cases prove that the technique is realizable and has interesting value for practical applications. The main characteristics are the simplicity of its implementation (just a few lines of code) and the ability to impose a desired dynamics in a collision situation. Furthermore, the feature to adapt the speed of reference in shared path situations is also remarkable.

In future works it is expected to implement this technique in real robots in order to have a complete validation. Another aspect that require more analysis is to bound the parameters involved in the operation of the technique to derive the sufficient condition for the SM establishment. A way to bound the values of the involved parameters in order to ensure these conditions will be presented in the chapter 6.

The work developed in this chapter has led to the presentation of two conference papers [54] (with its publication in IEEE Xplore), [55] and a journal publication in progress [56].

CHAPITRE 5

# ROBUST PID ROBOTIC CONTROL

In the previous chapters, a couple of techniques were developed to deal with signal constraints problems (both at the input and the output) which affect robotics systems. The performance of the previous auxiliary-loop techniques relies on the tune of a main controller. In this chapter a tuning method for the main controller is analized, which reinforces the robustness of the constraint compensating algorithms under structural constraints on the controller (see section 2.5). The controller is considered predefined as frequently happens in industrial or commercial robots. Particularly, the PID structure will be taken as a system constraint and a robust tuning technique will be addressed considering it.

The tunning method is used here to compute a controller for the AUV Ciscrea subjected to external disturbances. The control law's design objectives are formulated as $H_\infty$ objectives used to synthesize a robust controller. Then, a robustness analysis to AUV model uncertainties is performed without conservatism with interval analysis and global optimization in order to validate the control law. Some advantages of the approach are illustrated by comparing it with two other classical design methods by both simulations and experiments.

## 5.1 Robust PID problem

Beyond the choice of control structure, systems are subject to environmental disturbances. Examples of this type of disturbances can be marine current for AUV, adverse weather for aerial operations or a slippery surface for a land vehicle. Furthermore, not only disturbances represent a problem in robotic systems, but also the fact that in general the models are nonlinear and not perfectly known. These two problematics lead to the need for the use of robust control techniques, which in the design stages must establish a range of confidence in function of these non-idealities.

Many robust control techniques have been developed ($H_2$, $H_1$ and $H\infty - synthesis$), but most of them have the disadvantage of requiring system linearization and leading to elevated controller orders. Case of this are works like [57] where a synthesizing method for $H_\infty$ controller via singular value truncation is proposed, or [58] which focuses on the design of a robust multiple-input multiple-output $H_\infty$ controller to deal with a time-varying model. These works suffer from two disadvantages of the traditional $H_\infty$ solving method, the high order of the controller and the lack of robustness with respect to model uncertain parameters. Furthermore, another problem is added when a restriction on the controller's form (e.g. the proportional-integral-derivative (PID) structure) is imposed. This does not result in an easy-to-tackle mathematical problem in these formulations.

Although the structure controller restriction can be seen very restrictive, for example at the industrial level about 90$\%$ of the controls obey to the PID structure, and they are implemented at the low level of control (generally inaccessible) [59]. The control of the transient and steady-state response has made the PID control one of the most used controllers, offering a simple and efficient solution to several problems in the real world.

From the beginning different methods for PID tuning have been proposed, among which the most prominent has been Zigler-Nichols (ZN), since 1942 [60]. Nevertheless, ZN methods might be inadequate in applications where high performance is required because much a priori information of the processes is not exploited in the PID controller design. To tackle this problem, varieties of new techniques have been developed. Among them are the analytical tuning method [61] [60], the optimization based method [62] [63], the gain and phase margin method [64][65], and so forth.

A possible alternative to these solutions, looking for the goodness of classic robust control, is the use of interval techniques, suitable tools for the handling of parametric uncertainties.

The aim of this chapter is to implement a controller robust against both model uncertainties and external disturbances using an approach based on $H_\infty$ synthesis, constrained to a PID structure. The principle of the controller design is based on the combination of the interval arithmetic with a linear control theory [66] [67].

There are two main reasons to choose the $H_\infty$ approach : (i) $H_\infty$ synthesis enables to take multiple design constraints into account and (ii) robustness analysis against model uncertainties can be performed with respect to the $H_\infty$ objectives. Here, it is proposed to use the Matlab's Systune toolbox which enables synthesizing structured

controllers from $H_\infty$ specifications and also to perform a robustness procedure to take model uncertainties into account [68]. However, this procedure cannot ensure, in a guaranteed way, that the design and robustness constraints are reached for all possible values of model uncertain parameters.

The sensitivity analysis of design objectives over model uncertainty is a non-convex problem. In order to solve this problem, here it is proposed to use a global optimization approach which enables performing a robustness analysis in a guaranteed way based on Interval Analysis (see [69] and [70]).

## 5.2 Robust design procedure

This section presents a quick introduction to the $H_\infty$ problem, and the robustness analysis.

### 5.2.1 $H_\infty$ synthesis

Based on [71], $H_\infty$ synthesis is an method to design controllers from frequency specifications. The classical regulation scheme, considered for $H_\infty$ synthesis, is represented in Fig. 5.1, where $K$ is the controller to compute and $P$ is the plant to control. Both $P$ and $K$ are Linear Time Invariant (LTI) systems. In Fig. 5.1, $w$ represents the

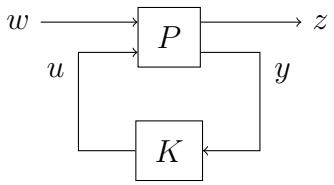



FIGURE 5.1 – $H_\infty$ synthesis classical regulation scheme.

vector of exogenous or perturbation inputs, $z$ the vector of performance outputs, $u$ the control signal and $y$ the measured outputs.
Let $F(P, K)$ be the Linear Fractional Transform of $P$ and $K$, which maps $w$ into $z$. $z = F(P, K)w$. We recall that the $H_\infty$ norm of an LTI plant is defined by (5.1), where $\sigma_{max}$ is the maximum singular value, $F(P, K)^*$ is the hermitian transpose of $F(P, K)$, $\omega$

is the pulsation in $rad/s$ and $i$ the imaginary unit.

$$\|F(P,K)\|_\infty = \sup_{\omega>0} \sigma_{max}\left(F(P,K,i\omega)^* \cdot F(P,K,i\omega)\right) \tag{5.1}$$

The $H_\infty$ synthesis aims to compute a controller that minimizes the $H_\infty$ norm of $F(P,K)$ and internally stabilizes the closed-loop system. To do so, the following problem is solved :

$$\begin{cases} \min\limits_{K} & \|F(P,K)\|_\infty \\ \text{subject to} & K \text{ stabilizes } F(P,K). \end{cases} \tag{5.2}$$

From a practical point of view, the $H_\infty$ synthesis computes a controller that minimizes the maximal response of the outputs $z$ to inputs $w$.
$P$ is an augmented plant built from $G$ the model of plant to be controlled, and filters that amplify the non-desired behaviors of the objective outputs $\tilde{z}$. $z$ is the weighted counterpart of the outputs $\tilde{z}$, $z = W\tilde{z}$ with $W$ a weighting filter. If both $w$ and $z$ are of dimension one, the $H_\infty$ norm corresponds to the maximum modulus of the transfer function from $w$ to $z$, denoted $T_{w\to z}$, over the pulsations. Then :

$$\begin{aligned} &\|WT_{w\to\tilde{z}}\|_\infty \leq 1 \iff \sup_{\omega>0} |W(i\omega)T_{w\to\tilde{z}}(i\omega)| \leq 1, \\ &\iff \forall\omega > 0,\ |T_{w\to\tilde{z}}(i\omega)| \leq |W^{-1}(i\omega)|. \end{aligned} \tag{5.3}$$

From (5.3), $W^{-1}$ can be interpreted as a frequency template that bounds the frequency response of $T_{w\to z}$.

The $H_\infty$ synthesis allows taking multiple objectives into account, such as minimization of tracking error, disturbance rejection, etc. Moreover, recent researches have been conducted to synthesize structured controller ([72], [73] and [70]).These methods propose to solve Problem (5.2) subject to a priori constraints on the controller, for example a PID structure constraint.

The last key point is that guaranteed robustness analysis can be performed on the $H_\infty$ norm of a system that suffers from model uncertainties, as it is explained in the following section.

### 5.2.2 Robustness analysis

In most real life applications, the model of the system to control suffers from uncertainties. These uncertainties may come from linear approximations or unknown values

of physical parameters of the system, for example. They can be taken into account either directly in the synthesis process, or after the synthesis of a controller performed from a nominal model by verifying that this controller ensures the performances for every possible value of the uncertainty. This section is focus on the robustness analysis of a controller synthesized for a nominal model with respect to model uncertainty.

Let $G(\sigma)$ be an LTI system which depends on real uncertain parameters $\sigma \in \Sigma$, where $\Sigma$ denotes the set of admissible values of the uncertainties. Suppose that a controller $K$ was synthesized for a nominal plant $G(\sigma_n)$, where $\sigma_n \in \Sigma$ is the central value of uncertainty, from constraints of the kind $\mathcal{C}(G, K) \leq 0$. The synthesis constraints $\mathcal{C}$ correspond in this case to stability constraints and $H_\infty$ constraints. Thus, $K$ is a solution to the problem (5.4).

$$\text{find } K \text{ such that } \mathcal{C}(G(\sigma_n), K) \leq 0 \tag{5.4}$$

The proposed robustness analysis consists in verifying that the constraints are satisfied for all values of uncertainties :

$$\text{Prove that } \mathcal{C}(G(\sigma), K) \leq 0,\ \forall \sigma \in \Sigma \tag{5.5}$$

In order to solve the problem (5.5), a global optimization approach based on Interval Analysis ([70] and [69]) is proposed. This is based on the use of interval arithmetic and the branch and bound algorithm. Briefly, this method starts from an initial interval of the parameter domain and begins to divide it successively. In each new division, it is checked if the analysis condition is fulfilled or not, ending when the desired resolution is reached. For the sake of clarity in the exposition, a detailed explanation of interval techniques as well as auxiliary methods will be carried out in chapter 6.

Interval Analysis combined with branch-and-bound algorithm can provide a guaranteed enclosure $[\underline{\mathcal{C}}, \overline{\mathcal{C}}]$ of $sup_{\sigma \in \Sigma}\mathcal{C}$, the maximum of $\mathcal{C}$ over $\Sigma$. This corresponds to the worst case among uncertainties. The problem (5.5) is not trivial in the general case, because functions $\mathcal{C}$ are non-convex. Indeed, the stability constraint can be formulated as several polynomial inequalities $R_i(\sigma) \leq 0$ using the Routh-Hurwitz criterion (see [74]), and the $H_\infty$ constraints as the modulus of a transfer $T$, $|T(\sigma, i\omega)| - 1 \leq 0$ (see (5.3)).

According to (5.6), if $\overline{\mathcal{C}} \leq 0$, it proves that the constraints are satisfied for all uncertainties and that $K$ is robust with respect to the model uncertainties. On the opposite,

if $\underline{C} > 0$, it proves that there exists at least one value of $\sigma$ that does not satisfy a constraint.

$$\overline{\mathcal{C}} \leq 0 \implies \forall \sigma \in \Sigma, \mathcal{C}(G(\sigma), K) \leq 0. \tag{5.6}$$

Using the global optimization algorithm to solve Problem (5.7), it is possible to prove in a guaranteed way whether or not stability constraint and $H_\infty$ constraints are respected for all possible values of $\sigma$ :

$$\sup_{\sigma \in \Sigma, \omega \in \Omega} \mathcal{C}(G(\sigma, i\omega), K(i\omega)) \tag{5.7}$$

where $\Omega$ is a bounded interval of $\mathbb{R}^+$ ([70]).

**Remark 1.** *A global optimization approach to robustness analysis of $H_\infty$ constraints presents an advantage compared to the classical $\mu$-analysis [75]. Indeed, $\mu$-analysis allows computing an upper bound of the frequency response over a finite number of pulsations, whereas global optimization provides an upper bound over all the pulsations in a bounded domain. As a consequence the robustness analysis proposed here gives a reliable guarantee that $\mu$-analysis is unable to provide.*

## 5.3 Application case : AUV Ciscrea yaw control

The design of control laws for AUV presents three main problems :

— the non-linear dynamics of the vehicle,

— the model uncertainties resulting from the non-exact knowledge of the hydrodynamic coefficients,

— the external disturbance of the environment.

In this section, it is proposed to synthesize a controller to control the yaw direction of the Ciscrea robot modeled in section 3.2. The control scheme is given by Fig. 5.2, where $r$ is the reference signal, $e$ the error signal, $u$ the control signal, $d$ the disturbance input and $\psi$ the measure of the yaw angle. The control law must ensure a small tracking error and should not be sensitive to external disturbances. To do so, $H_\infty$ constraints are defined for a linear model of the yaw behavior of the robot, and the controller is synthesized from these constraints.

The equation that describes the yaw angle dynamics of the Ciscrea is given by the last row of model (3.11). This row, due to the low coupling between movement directions,

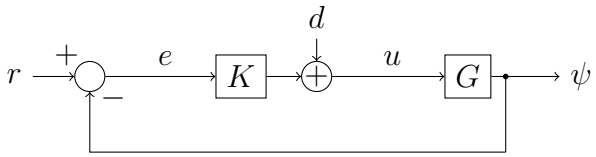



FIGURE 5.2 – Control scheme

can be considered independent of the rest of the model. Since $H_\infty$ synthesis can be applied only on LTI systems, the non-linear dynamics of Ciscrea yaw direction, the non-linear behaviour of actuators (Eq. 3.12) and a compass delay of 0.5 seconds are linearized to provide the transfer function :

$$\frac{\psi(s)}{r(s)} = \frac{0.3931}{s^2 + 2.08\delta s}\frac{1 - 0.25s}{1 + 0.25s}. \tag{5.8}$$

The first rational factor corresponds to the yaw dynamics, where $\delta$ is the yaw angular velocity at which the system is linearized. Its value can vary between 0 and 4 rad/s. The second rational factor corresponds to a first order Pade approximation of the delay. As a consequence, the yaw dynamics is approximated by a family of linear systems resulting from the linearization at different velocities.

The objective is to control the yaw angle attending to the following criteria :

— The error between the AUV yaw angle and the desired yaw angle must be small.
— The AUV must not be sensitive to torque perturbations due to the environment.
— The control structure is fixed as a filtered Proportional Integral Derivative (PID) controller.

These lead to the synthesis problem (5.9), where if the norms are under $1$, then the specifications are guaranteed.

$$\begin{array}{l} \text{Find } K \text{ such as } \alpha \text{ is minimum} \\ \left\{ \begin{array}{l} \|W_e T_{r\to e}\|_\infty \leq \alpha, \\ \|W_e T_{d\to e} W_d\|_\infty \leq \alpha, \\ \|W_u T_{r\to u}\|_\infty \leq \alpha, \\ K \text{stabilizes the closed-loop system.} \end{array} \right. \end{array} \tag{5.9}$$

with

$$W_e(s) = \tfrac{0.1s+0.6283}{s+0.06283}, \quad W_d(s) = \tfrac{0.1s+0.06283}{s+0.6283}, \quad W_u = 0.167.$$

These criteria can be translated as a small sensitivity of the error signal to the reference and the disturbance input. More precisely, we want the sensitivity to be small in the frequency domain where the robot behaves, that is in the pulsation domain $[0, \omega_c]$, where $\omega_c = 1\ rad/s$ is the cut-off frequency of the closed-loop with negative unitary feedback of the linear model given by (5.8). This point leads to the $W_e$ shaping. In addition, we suppose that the spectrum of external disturbances is located in $[0, 0.1]$ Hertz. This point leads to the $W_d$ shaping. Finally, we want to limit the control signal in order to avoid actuator saturation. This point leads to the shaping of $W_u$.

**Remark 2.** *Note that in problem (5.9) the focus is on the robustness against external perturbations. Nevertheless, it would be possible to add in the optimization problem statement (5.9) extra conditions to match additional robustness features. For example, considering multiplicative uncertainty.*

Given the constraint on the controller structure, it is proposed to synthesize a PID controller with a particular plant $G(\tilde{\delta})$, with $\delta = \tilde{\delta} = 2$. This choice is justified by the trade off between no damping (that leads to very low control command) and (high damping that leads to very high control command). The PID controller has the form : $K(k, s) = k_p + \frac{k_i}{s} + \frac{k_d s}{1+Ts}$ with $k = (k_p, k_i, k_d, T)$. Thus, both transfer functions $T_{r\to e}(k, i\omega)$ and $T_{d\to e}(k, i\omega)$ depend on $k$. The Matlab's toolbox Systune provides the following solution :

$$\tilde{k} = (4.68, 0.71, 4.68, 0.11).$$

The control law is robust if both stability and $H_\infty$ constraints are accomplished for all $\delta \in [0, 4]$. This can be proved as true or false in a guaranteed way using interval arithmetic as explained in subsection 5.2.2. The stability of the closed-loop system can be expressed as a set of four polynomial inequalities with the Routh-Hurwtiz criterion. Using the algorithm based on Interval Analysis, the robustness analysis of the stability constraints provides the following upper bound :

$$\sup_{\delta\in[0,4]} R_i(\delta, \tilde{k}) \leq -0.01, \forall i \in \{1, ..., 4\},$$

which proves that $K(\tilde{k})$ robustly stabilizes the linear closed-loop system. Indeed, the closed-loop system is stable with the controller $\tilde{k}$ for all $\delta \in [0, 4]$.
Moreover, the robustness analysis of $H_\infty$ constraints over the pulsation range $[0, \omega_c]$

provides the following results :

$$\sup_{\delta\in[0,4]} \{\|W_e T_{r\to e}(\tilde{k})\|_\infty\} \in [6.55, 7.20]$$
$$\sup_{\delta\in[0,4]} \{\|W_e T_{d\to e}(\tilde{k}) W_d\|_\infty\} \leq 0.56$$
$$\sup_{\delta\in[0,4]} \{\|W_u T_{r\to u}(\tilde{k})\|_\infty\} \leq 0.89$$

As a consequence, we conclude that one out of three $H_\infty$ constraints is not achieved for some values of $\delta$. In order to know for which pulsations and uncertain parameters the frequency template $W_e^{-1}$ is over-passed by the transfer $T_{r\to e}$, the frequency response is plotted in Figure 5.3 for ten values of the uncertainty $\delta$.

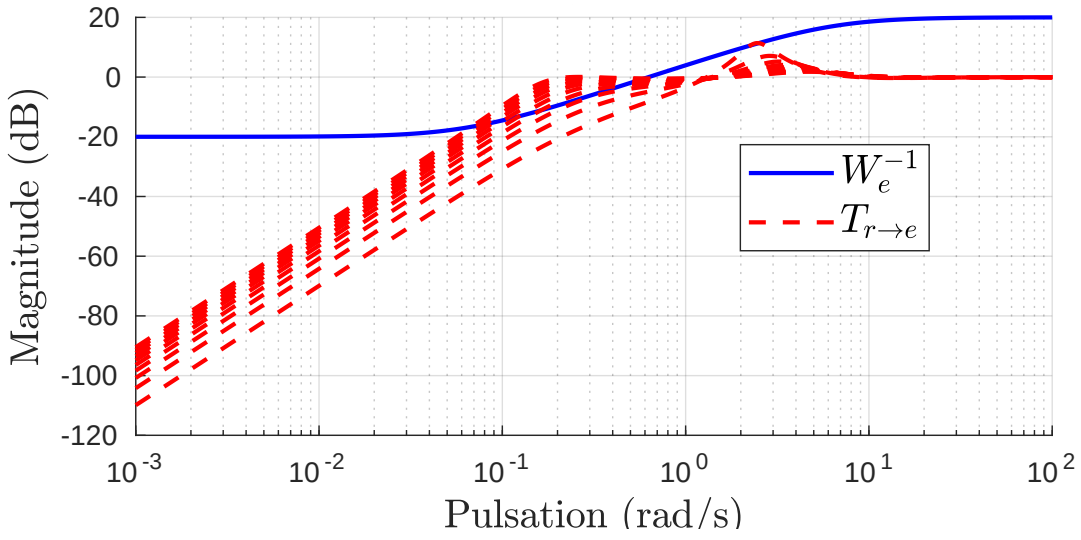



FIGURE 5.3 – Bode diagram of $T_{r\to e}$ for diferrent $\delta$ values.

It appears in Figure 5.3 that the closed-loop system does not have the required performance in the pulsation range $[10^{-1.6}, 10^{-0.4}]$ for all values of $\delta$. Over this pulsation range, the gain of the response increases with respect to the value of $\delta$, which means that the frequency template $W_e^{-1}$ is not satisfied but for high angular velocities only.

When using $H_\infty$ synthesis, if the $H_\infty$ constraints are not respected the general procedure is to modify the weighting functions to be less demanding with respect to the closed-loop system performances, until a controller is found such that the constraints are fulfilled, i.e. that the performances are guaranteed.

In this case, even if $K(\tilde{k})$ does not respect the error $H_\infty$ constraints for some values

of $\delta$, the study of the frequency response of the closed-loop system shows that the controller has acceptable performances. In addition, a robust stability analysis enables to guarantee the stability of the linear system, which makes $K(\tilde{k})$ a potentially good controller. Its performance must be validated by simulations with the non-linear model and by experiments. In order to compare the performance of the designed controller, two other PID designs are used. First, a controller tuned using the Ziegler-Nichols frequency response method, denoted as ZN controller : $k_{ZN} = (1.32, 0.22, 1.89, 0.5)$, which main design criterion is to obtain a quarter amplitude decay ratio for the load disturbance response. Second, another controller tuned according to the rules exposed in [60], denoted as the Chien controller, for a linear system in the form of (5.8) and a value of $\delta = 2$ : $k_{Chien} = (1.82, 0.12, 6.4, 0.35)$.

## 5.4 Results

In this section, the three controllers are compared over simulations and real experiments. The main objective is to show the robustness of the proposed controller (denoted as Hinf) against perturbations and nonlinearities.

### 5.4.1 Simulations

The simulations were done using the non-linear model described in section 3.2. Three simulations are presented : a step response, a response to a constant perturbation, and a response to a random perturbation.

The first simulation presents the step response of the system. The objective is to compare the overpass and settling time. In Fig. 5.4 this can be appreciated. In this figure, the Hinf controller has a higher overpass than Chien controller, but at the same time the settling time is shorter. This overpass is a consequence of the $H_\infty$ tuning, and actually, this was not considered as a constraint in the design problem.

The second simulation consists in the application of a step perturbation over the control input, filtered by $W_d$, after a long period in which the system is regulating its bow. Here it is sought to anticipate the response of the system to sudden changes in marine currents, within the frequency range of design. Figure 5.5 shows this situation highlighting only the response of the system to the step perturbation applied at $t = 100$s.

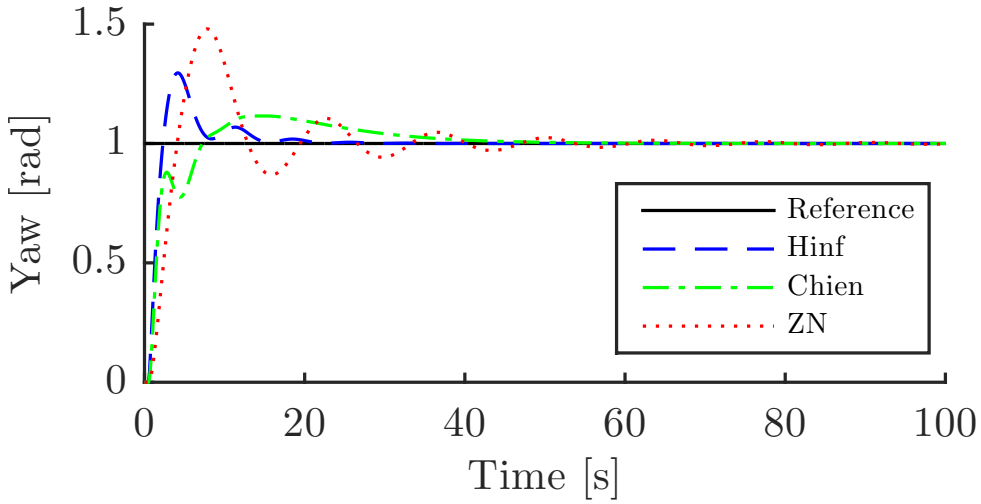



FIGURE 5.4 – Step response simulation

It is possible to observe that the Hinf controller rejects this perturbation well, contrary to ZN and Chien controllers. Table 5.1 presents the Root-Mean-Square Error (RMSE), the Normalized Mean Absolute Error (NMAE) and bias of this simulations, where the numerical values obtained confirm the improvement in the response by the Hinf controller in comparison to the others.

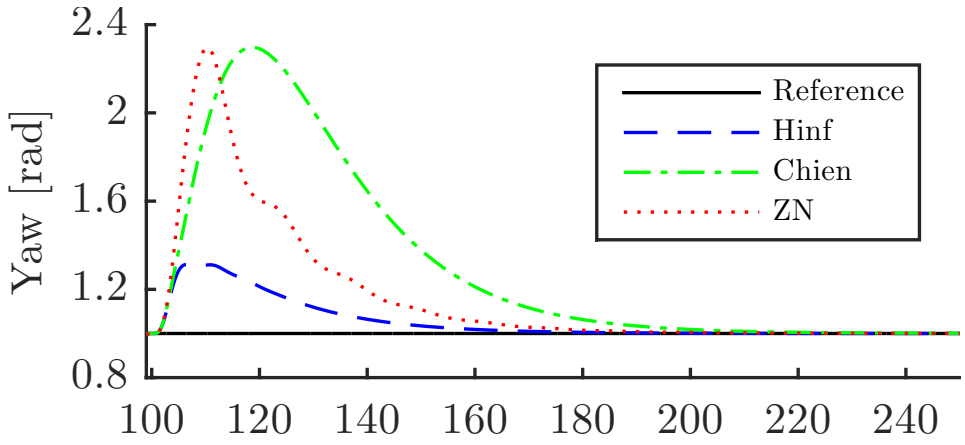



FIGURE 5.5 – Simulation of step perturbation at $t = 100$s

TABLE 5.1 – Simulation step perturbation errors

| Simulation | RMSE | NMAE | BIAS |
|---|---|---|---|
| ZN | 0.3596 | 0.0944 | -0.1627 |
| Hinf | 0.1090 | 0.0425 | -0.0519 |
| Chien | 0.5177 | 0.1269 | -0.2557 |

In the last simulation a white uniform noise signal filtered by the weighting function $W_d$ (in this way the system is excited in the bandwidth where the disturbances are expected) is applied as a disturbance to the control input. This simulation demonstrates the response of the system to random marine currents, Fig. 5.6 shows the yaw output. The Hinf controller has a remarkable performance in these conditions. Table 5.2 provides the errors of this simulation in order to have numerical values of the performance improvement.

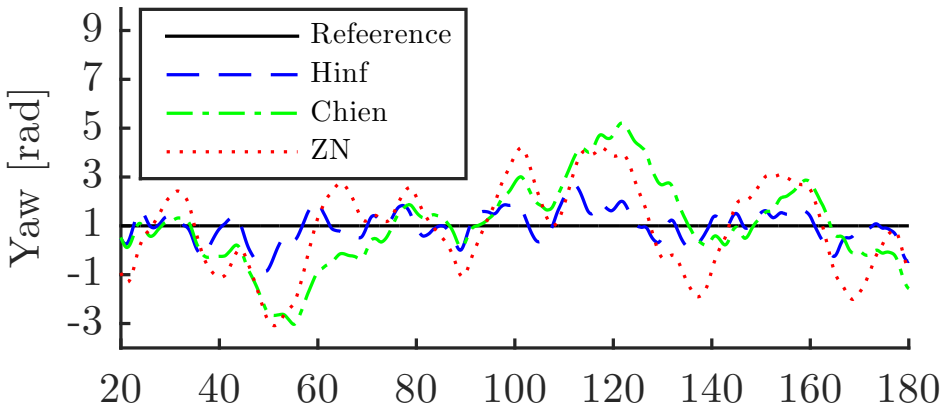



FIGURE 5.6 – Filtered white noise perturbation simulation

TABLE 5.2 – Simulation filtered white noise perturbation errors

| Simulation | RMSE | NMAE | Bias |
|---|---|---|---|
| ZN | 1.7132 | 2.7969 | -0.1263 |
| Hinf | 0.6017 | 1.2138 | -0.0480 |
| Chien | 1.6816 | 2.6120 | -0.1334 |

### 5.4.2 Experimental Results

The three controllers are compared over three experiments conducted at the ENSTA-Bretagne facilities. Each experiment consists in testing the performance of the three controllers on the real robot subject to perturbations. In all the cases, the perturbation was generated by an external 12V propeller with a constant rotational speed. Since the currents generated by the propeller are not constant, these perturbations will behave similar to the random disturbance presented in simulation.

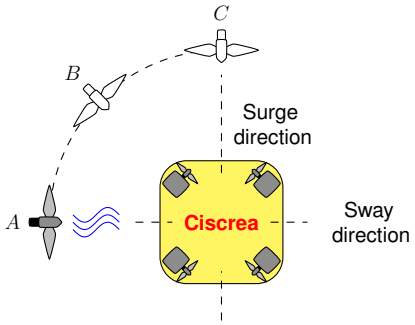



FIGURE 5.7 – Top view of experiment setup

The first experiment consists in undergoing the AUV to an external perturbation aligned to its sway direction (see A in Fig. 5.7). In Figure 5.8, the yaw measurement is displayed for each controller. In this first test, the unperturbed experimental step response is shown to compare it with simulation results (it can be appreciated from 0 to 40s). At 40s, the external perturbation is applied.

From this figure, we can observe the same behaviour as the one predicted by the simulation referred to the step response, and a good rejection of the perturbation for all the techniques employed. The errors in the trials are listed in Table 5.3, where all the simulation is considered (even the step response). As this experiment tends little to destabilize the bow of the AUV, we see that the obtained performance for the perturbation rejection is similar for all controllers.

TABLE 5.3 – Errors : Perturbation in sway direction

| Experiment | RMSE | NMAE | BIAS |
|---|---|---|---|
| ZN | 0.2166 | 0.0689 | -0.0204 |
| Hinf | 0.1355 | 0.0386 | -0.0230 |
| Chien | 0.1738 | 0.0762 | -0.0137 |

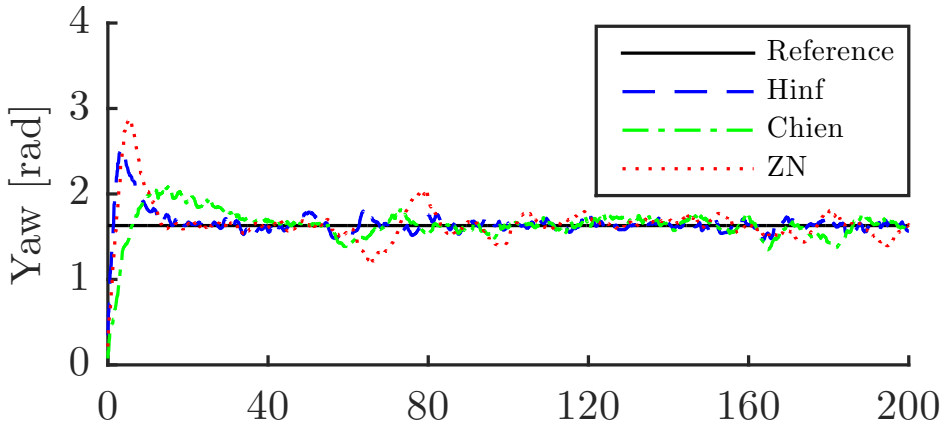



FIGURE 5.8 – Yaw measurements when facing a step response at $t = 0$s and a perturbation in sway direction at $t = 40$s

The second experiment consists in exposing the AUV to a perturbation at 45° of sway direction (see B in Fig. 5.7). The perturbation is applied from the beginning of the experiment. The response of the system is shown in Fig. 5.9. From Table 5.4, where a comparison of the errors between the different controllers is presented, it is possible to conclude that the Hinf controller shows a satisfactory performance and a good rejection to disturbances for this situation.

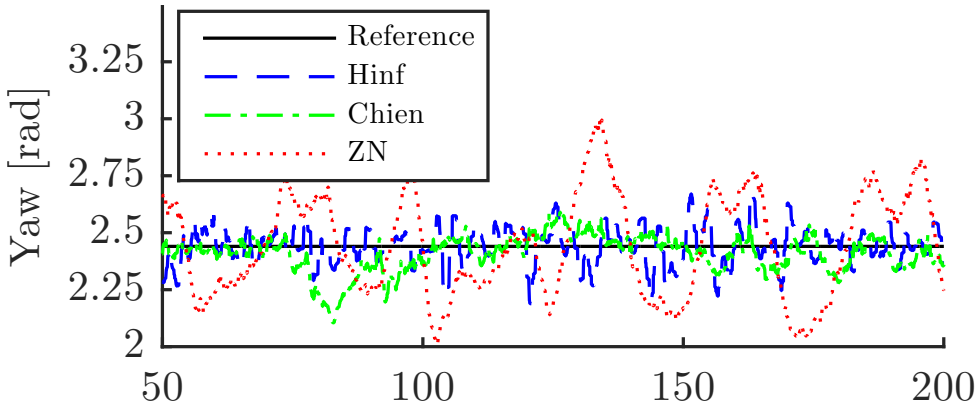



FIGURE 5.9 – Yaw measurements when facing a perturbation at 45° of surge direction

The last experience consists in applying a perturbation in the surge direction (see C in Fig. 5.7). As above, perturbation is applied from the beginning of the experiment and results are shown in Fig. 5.10. Since the direction of the AUV's bow coincides with the direction of disturbance, the latter has a greater effect. The errors of this experiment

TABLE 5.4 – Errors : Perturbation at 45° of surge direction

| Experiment | RMSE | NMAE | Bias |
|---|---|---|---|
| ZN | 0.1742 | 0.0502 | 0.0137 |
| Hinf | 0.0650 | 0.0174 | 0.0037 |
| Chien | 0.0755 | 0.0179 | 0.0172 |

are provided in Table 5.5. As can be appreciated the Hinf controller is the one which better rejects this disturbance.

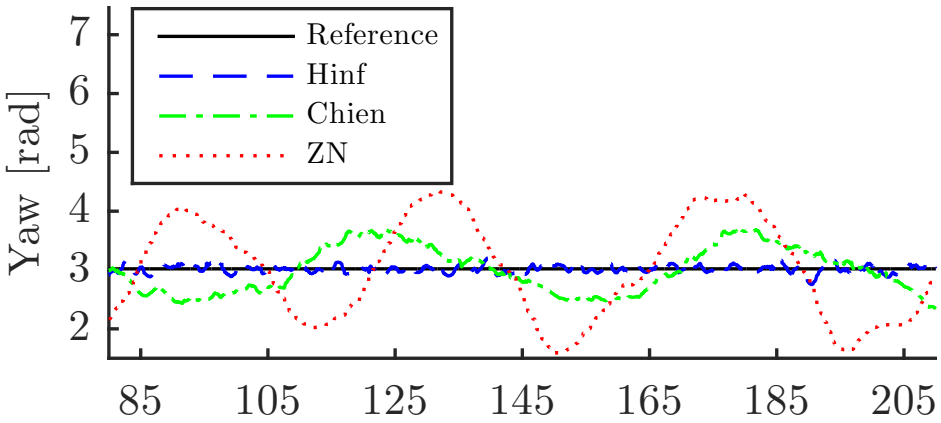



FIGURE 5.10 – Yaw measurements when facing a perturbation in surge direction

TABLE 5.5 – Errors : Perturbation in surge direction

| Experiment | RMSE | NMAE | Bias |
|---|---|---|---|
| ZN | 0.3957 | 0.0749 | -0.0037 |
| Hinf | 0.0371 | 0.0059 | -7.1612e-04 |
| Chien | 0.2548 | 0.0482 | 0.0256 |

Overall, from the three experiments, it can be concluded that the Hinf PID controller design has an acceptable performance in real scenarios despite its structural constraints, model uncertainties (recall it was designed from the linear model $G(\tilde{\delta})$) and external disturbances.

## 5.5 Conclusions

This chapter proposed a methodology for the analysis and design of a robust structured control law. As has been shown the use of techniques with structured constraints has relevance for applications with already implemented controllers.

The $H_\infty$ synthesis allows computing a controller which takes three constraints into account at the same time : small tracking error, low sensitivity to external disturbances and saturation avoidance of actuators. A robustness analysis with global optimization tools based on interval analysis enables to analyze which design constraints are reached and to ensure stability over a continuous set of operating angular velocities.

The comparison of the controller designed from $H_\infty$ constraints with two other controllers obtained from empirical methods, widely used in numerous applications, emphasized the advantages of the approach. It highlights the superior experimental results of the proposed technique, in accordance with the results of simulation. The work developed in this chapter has led to the presentation of two conference papers [76] and [77], and its publication in IFAC-Papers-OnLine.

CHAPITRE 6

# GLOBAL OPTIMIZATION FOR NONLINEAR ROBOTIC CONTROL

In chapter 3 and 4, input and output constraint compensating algorithms were presented. In chapter 5 a robust way to tune a main structured-constrained controller was described. In this chapter a way to integrate the previous ideas under global optimization considerations to deal with both uncertainties and nonlinearity constraints is exposed.

The classical SM design can lead to over or underestimation of the sliding domain, closed-loop robustness and necessary control power. Here, the design of SM is addressed from a global optimization approach using interval arithmetic. A solution to the analysis and synthesis problems of SM is provided, where the necessary and sufficient conditions are fulfilled in a guaranteed way.

The proposals covered in this chapter will provide robust control even for non-linear systems. At the same time, analytical tools will be developed to certify compliance with the sufficient conditions of SM, thus completing the tuning of proposals in chapters 3 and 4.

For the analysis problem the proposed methodology allows checking sliding mode behaviour over given state domain and parameter sets, generating a series of maps called subpavings. For the synthesis problem, the methodology allows designing the sliding manifold and switching logic with a given optimization criterion. The methodology is illustrated via its application to the Ciscrea AUV heave axis control.

## 6.1 SM design problem

This section presents a review of the theoretical aspects of SM, global optimization and interval analysis. The link between these three items will be done in section 6.2.

### 6.1.1 SM control theory : the equivalent control signal

Here, a fast review of SM theory, exposed in the subsection 3.3.1, is made. Looking back to this, the SM allows imposing a prescribed closed-loop dynamics to a system (Eq. 3.13) by applying a discontinuous control action (Eq. 3.14). This action can take two values following a discontinuous control law with an associated manifold on the state-space (sliding surface). The objective is to force the system to reach the desired sliding surface and then to slide on it though a very fast switching action [44].

Among other attractive features sliding regimes are easy to implement, reduce the order of the system dynamics, and provide robustness to matched uncertainties and external disturbances.

The design procedure consists of two stages. First, the equation of the manifold where the system slides is selected in accordance with some performance criterion for the desired dynamics. Then, the discontinuous control should be found such that the system states reach the manifold and SM exists on this manifold. This desired dynamics must be compatible with the physical limits of our system, later we will return to this point through the analysis procedure proposed in this chapter.

Returning to subsection 3.3.1, the sliding surface $S$ is defined as the manifold where the auxiliary output, also called switching function, vanishes :

$$S = \{\mathbf{x} \in \mathbb{R}^n \mid \sigma(\mathbf{x}) = \mathbf{0}\}\,. \tag{6.1}$$

If as a result of the switching policy Eq. 3.14, the reaching condition

$$\begin{cases} \dot{\sigma}(\mathbf{x}) < \mathbf{0} & if \quad \sigma(\mathbf{x}) > \mathbf{0} \\ \dot{\sigma}(\mathbf{x}) > \mathbf{0} & if \quad \sigma(\mathbf{x}) < \mathbf{0} \end{cases} \tag{6.2}$$

locally holds on both sides of the surface, a switching sequence of very high frequency (ideally infinite) occurs, constraining the system state trajectory to slide on S.

For sliding motion to exist on $S$ (i.e. for satisfying condition (6.2)), the auxiliary output $\sigma(\mathbf{x})$ must have unitary relative degree with respect to the discontinuous signal, i.e. its first derivative must explicitly depend on $u$ [44].

For the proposal of this chapter, it results interesting to define the ideal SM using the equivalent control concept. Taking the invariant conditions over the SM surface, we

get :

$$\begin{cases} \sigma(\mathbf{x}) = 0 \\ \dot{\sigma}(\mathbf{x}) = \frac{d\sigma}{d\mathbf{x}}\dot{\mathbf{x}} = L_{f+gu_{eq}}\sigma = L_f\sigma + L_g\sigma u_{eq} = 0 \end{cases} \tag{6.3}$$

where the generic operator $L_f h(x) : \mathbb{R}^n \to \mathbb{R}$ (directional or Lie derivative) denotes the derivative of a scalar field $h(x) : \mathbb{R}^n \to \mathbb{R}$ in the direction of a vector field $f(x) : \mathbb{R}^n \to \mathbb{R}^n$

$$L_f h(x) = \frac{\partial h}{\partial x} f(x). \tag{6.4}$$

From (6.3) it is possible to obtain $u_{eq}(\mathbf{x})$, a smooth control law which makes $S$ an invariant subset.

$$u_{eq}(\mathbf{x}) = -\frac{L_f\sigma}{L_g\sigma} \tag{6.5}$$

Following this approach it is possible to arrive to the necessary and sufficient condition for the SM. It is observed in (6.5) that $L_g\sigma \neq 0$ is necessary for the existence of $u_{eq}$ and, therefore of SM. Furthermore, a necessary and sufficient condition for the local existence of the SM over $S$ can be derived from (6.2) and (6.3). If we consider (without loss of generality) $u^+ > u^-$ it must hold :

$$u^-(\mathbf{x}) < u_{eq}(\mathbf{x}) < u^+(\mathbf{x}) \tag{6.6}$$

From (6.6), $u_{eq}(\mathbf{x})$ can be interpreted as an average control action between the maximal and minimal discontinuous action of the system.

### 6.1.2 Global optimization (GO)

Let us consider a continuous constrained optimization problem formulated as :

$$\begin{cases} \inf\limits_{\mathbf{k}\in\mathbb{R}^n} & m(\mathbf{k}) \\ \text{subject to} & c(\mathbf{k}) \leq 0, \end{cases} \tag{6.7}$$

where $m$ is the objective function which maps $\mathbb{R}^n$ into $\mathbb{R}$, $\mathbf{k} \in \mathbb{R}^n$ is the optimization variable, and $c$ is a function that maps $\mathbb{R}^n$ into $\mathbb{R}$ used to define a subset of $\mathbb{R}^n$ in which the solution is searched. The solution, also called the minimizer, is denoted as $\mathbf{k}^*$ and is the point where $m$ is minimum over the set defined by $\{\mathbf{k} \in \mathbb{R}^n, c(\mathbf{k}) \leq 0\}$. The minimum

is denoted as : $m^* = m(\mathbf{k}^*)$. From the definition of the minimum, property (6.8) holds :

$$\forall \mathbf{k} \in \mathbb{R}^n \text{ such as } c(\mathbf{k}) \leq 0,\ m(\mathbf{k}) \geq m^*. \tag{6.8}$$

If $m$ and $c$ are not convex functions, local optimization techniques have no warranty to converge to the global solution $\mathbf{k}^*$. On the other hand, global optimization methods converge to the global minimum and provide an enclosure $[\underline{m^*}, \overline{m^*}]$ of $m^*$. One well-known technique from global optimization is the Branch and Bound algorithm based on interval arithmetic [69].

### 6.1.3 Interval arithmetic

In order to present the Interval Branch and Bound Algorithm (IBBA), and the Set Inversion Via Interval Analysis (SIVIA) algorithm some definitions must be given [78].

**Definition 1.** *An interval $[k]$ is a closed connected subset of $\mathbb{R}$ [79], described by its endpoints $\underline{k}$ and $\overline{k}$ :*

$$[k] = [\underline{k}, \overline{k}] = \{k \mid \underline{k} \leq k \leq \overline{k}\}$$

*with $\underline{k} \in \mathbb{R} \cup \{-\infty\}$ and $\overline{k} \in \mathbb{R} \cup \{+\infty\}$*

The set of real intervals is denoted by $\mathbb{IR}$. A box $[\mathbf{k}]$ is an n-dimensional interval vector, and belongs to the space $\mathbb{IR}^n$.

**Definition 2.** *Let $[\mathbf{k}] \in \mathbb{IR}^n$ be a box. An inclusion function $[m]$ of $m$ maps $\mathbb{IR}^n$ into $\mathbb{IR}$ fulfills the following property :*

$$m([\mathbf{k}]) = \{m(k),\ k \in [\mathbf{k}]\} \subseteq [m]([\mathbf{k}])$$

Interval arithmetics extends common operators ($+$, $-$, $\times$, $sin$, $cos$, $\exp$, $\log$,...) to $\mathbb{IR}$ and provides inclusion functions of most of analytic functions.

Considering the problem (6.7), let us suppose that inclusion functions of $m$ and $c$ can be defined, and the minimizer $\mathbf{k}^*$ is searched in $\mathbb{K} \subset \mathbb{IR}^n$. The IBBA computes a guaranteed lower bound $\underline{m}$ and an upper bound $\overline{m}$ of $m^*$. To do so, IBBA repeatedly bisects $\mathbb{K}$ in smaller boxes $[\mathbf{k}_i]$ and discards them if it is proven that $\mathbf{k}^* \notin [\mathbf{k}_i]$. This

happens if the constraint is not satisfied over $[\mathbf{k}_i]$ :

$$\begin{aligned} \underline{[c]([\mathbf{k}_i])} > 0 \quad &\iff \forall \mathbf{k} \in [\mathbf{k}_i], c(\mathbf{k}) > 0, \\ &\implies \mathbf{k}^* \notin [\mathbf{k}_i], \end{aligned} \tag{6.9}$$

or if a feasible point $\tilde{k}$ has been found (through the testing of random points in each box) such that any points in $[\mathbf{k}_i]$ can provide a better feasible solution :

$$\underline{[m]([\mathbf{k}_i])} > m(\tilde{k}) \geq m^* \implies \mathbf{k}^* \notin [\mathbf{k}_i]. \tag{6.10}$$

The IBBA stops when the distance between $\underline{m}$ and $\overline{m}$ reaches the desired precision $\epsilon$, with

$$\underline{m} = \inf_i \underline{[m]([\mathbf{k}_i])}, \text{ and } \overline{m} = m(\tilde{k}) \tag{6.11}$$

Fig. 6.1 illustrates IBBA. The box $[\mathbf{k}_1]$ is proved not to contain $\mathbf{k}^*$ due to Property (6.9), as well as boxes $[\mathbf{k}_2]$ and $[\mathbf{k}_3]$ due to Property (6.10).

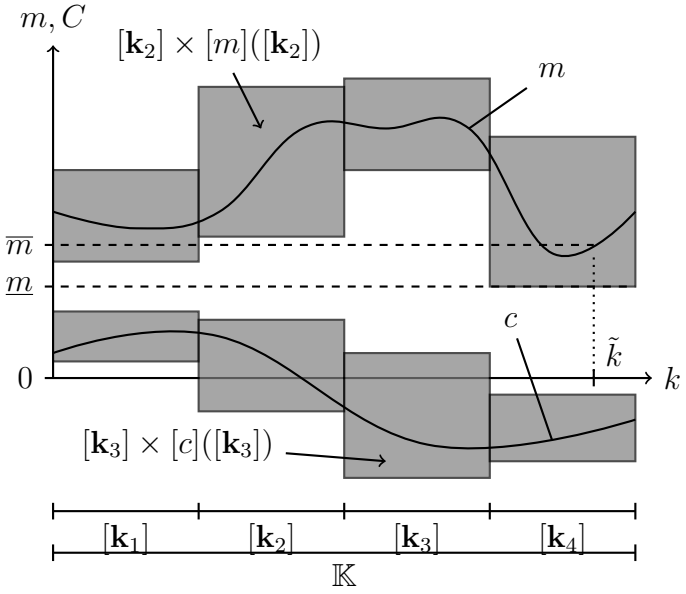



FIGURE 6.1 – Illustration of IBBA and SIVIA algorithms.

The SIVIA algorithm is a branch and bound method which allows approximating the feasible region of $\mathbb{K}$ through a subpaving, which is the union of non- overlapping boxes [74]. SIVIA algorithm bisects $\mathbb{K}$ in smaller boxes $[\mathbf{k}_i]$ until the constraint is proved to be

fulfilled over $[k]_i$ as a result of (6.12) or not to be fulfilled due to (6.9).

$$\overline{[c]([\mathbf{k}_i])} \leq 0 \iff \forall \mathbf{k} \in [\mathbf{k}_i], c(\mathbf{k}) \leq 0 \tag{6.12}$$

SIVIA algorithm stops when boxes $[\mathbf{k}_i]$ reach a minimum size $\epsilon$. In Fig. 6.1, SIVIA returns the subpaving composed of $[\mathbf{k}_1]$, $[\mathbf{k}_2]$, $[\mathbf{k}_3]$ and $[\mathbf{k}_4]$ indicating that $[\mathbf{k}_1]$ is not a subset of the feasible set, $[\mathbf{k}_4]$ is a subset of the feasible set, and that nothing could be proved for $[\mathbf{k}_2]$ and $[\mathbf{k}_3]$. That is, $[\mathbf{k}_4]$ is an inner approximation of the feasible set and $[\mathbf{k}_1] \cup [\mathbf{k}_2] \cup [\mathbf{k}_3]$ is an outer approximation. These approximations can be improved by bisecting $[\mathbf{k}_2]$ and $[\mathbf{k}_3]$ in smaller boxes.

Finally, IBBA has $[m]$, $[c]$, $\mathbb{K}$ and $\epsilon$ as inputs and provides a feasible point $\tilde{\mathbf{k}}$ and a guaranteed enclosure $[\underline{m}, \overline{m}]$ of the global minimum $m^*$. SIVIA algorithm has $[c]$, $\mathbb{K}$ and $\epsilon$ as inputs and provides a subpaving which characterizes the feasible region.

## 6.2 Proposal : SM design via GO techniques

In this section, the concepts discussed previously are integrated with the aim of solving problems of SM analysis and synthesis.

### 6.2.1 SM analysis problem

Given a desired sliding surface $\sigma(\mathbf{x}) = 0$ with $\mathbf{x}$ states of the system ($\sigma$ with relative degree one with respect to the discontinuous signal $u$), the analysis problem consists in verifying if $u_{eq}$ fulfills condition given by (6.6).

This problem can be addressed by solving the following program,

$$\left\{ \sup_{\boldsymbol{\delta} \in \Delta} \; a(\boldsymbol{\theta}, \boldsymbol{\delta}) \right. \tag{6.13}$$

with $\boldsymbol{\theta}$ being a vector of constant tuning parameters given by the operator (for example coefficients of the sliding surface), $\boldsymbol{\delta}$ the vector of variable parameters, $\Delta$ a subset of $\mathbb{IR}^{n_{\boldsymbol{\delta}}}$ with $n_{\boldsymbol{\delta}}$ the dimension of $\boldsymbol{\delta}$ and $a$ is the analysis function :

$$a = \max(u^- - u_{eq}(\boldsymbol{\theta}, \boldsymbol{\delta}), u_{eq}(\boldsymbol{\theta}, \boldsymbol{\delta}) - u^+). \tag{6.14}$$

Under these conditions, IBBA can provide an enclosure $[\underline{a}, \overline{a}]$ of the minimum $a^*$, and this can be used to ensure that the sliding condition holds over $\sigma(\mathbf{x}) = 0$.

From Property (6.8), we can derive Properties (6.15) and (6.16).

$$\begin{aligned}
&\overline{a} < 0 \\
&\implies a^* < 0 \\
&\iff \forall \boldsymbol{\delta} \in \Delta, \max(u^- - u_{eq}(\boldsymbol{\theta}, \boldsymbol{\delta}), u_{eq}(\boldsymbol{\theta}, \boldsymbol{\delta}) - u^+) < 0 \\
&\iff \forall \boldsymbol{\delta} \in \Delta, u^- - u_{eq}(\boldsymbol{\theta}, \boldsymbol{\delta}) < 0 \text{ and } u_{eq}(\boldsymbol{\theta}, \boldsymbol{\delta}) - u^+ < 0 \\
&\iff \forall \boldsymbol{\delta} \in \Delta, u^- < u_{eq}(\boldsymbol{\theta}, \boldsymbol{\delta}) \text{ and } u_{eq}(\boldsymbol{\theta}, \boldsymbol{\delta}) < u^+
\end{aligned} \tag{6.15}$$

Property (6.15) provides a sufficient condition to prove that the system will slide on the sliding surface $S$ over the subset $\Delta$.

$$\begin{aligned}
&\overline{a} \geq 0 \\
&\implies a^* = a(\boldsymbol{\delta}^*) >= 0 \\
&\iff \max(u^- - u_{eq}(\boldsymbol{\theta}, \boldsymbol{\delta}^*), u_{eq}(\boldsymbol{\theta}, \boldsymbol{\delta}^*) - u^+) \geq 0 \\
&\iff \exists \boldsymbol{\delta} \in \Delta, \max(u^- - u_{eq}(\boldsymbol{\theta}, \boldsymbol{\delta}^*), u_{eq}(\boldsymbol{\theta}, \boldsymbol{\delta}^*) - u^+) \geq 0 \\
&\iff \exists \boldsymbol{\delta} \in \Delta, u^- - u_{eq}(\boldsymbol{\theta}, \boldsymbol{\delta}) \geq 0 \text{ or } u_{eq}(\boldsymbol{\theta}, \boldsymbol{\delta}) - u^+ \geq 0 \\
&\iff \exists \boldsymbol{\delta} \in \Delta, u^- \geq u_{eq}(\boldsymbol{\theta}, \boldsymbol{\delta}) \text{ or } u_{eq}(\boldsymbol{\theta}, \boldsymbol{\delta}) \geq u^+
\end{aligned} \tag{6.16}$$

Property (6.16) provides a sufficient condition that the system will not slide over $S$ in all $\Delta$. Actually, the system will leave $S$ at least at $\boldsymbol{\delta}^*$ the solution to Problem (6.13). If $0 \in [\underline{a}, \overline{a}]$, it is not possible to prove whether or not $\boldsymbol{\theta}$ is a feasible solution since none of the conditions of Properties (6.15) and (6.16) is satisfied.

### 6.2.2 SM synthesis problem

Synthesis problems consist either in characterizing the set of feasible tuning parameters with respect to SM conditions and let the designer choose $\boldsymbol{\theta}$ in this set, or in minimizing a given cost function over this feasible set. SIVIA algorithm and IBBA are suited to perform such computation.

Let $\Theta$ be a subset of $\mathbb{IR}^{n_\theta}$, $l : \mathbb{R}^{n_\theta} \mapsto \mathbb{R}$ be a cost function given by the system designer. it is defined the analysis function at $\boldsymbol{\theta}$ by :

$$a_\theta = \max(u^- - u_{eq}(\boldsymbol{\theta}, \boldsymbol{\delta}), u_{eq}(\boldsymbol{\theta}, \boldsymbol{\delta}) - u^+) \tag{6.17}$$

and

$$a_\theta^* = \inf \boldsymbol{\delta} \in \Delta a_\theta(\boldsymbol{\delta}) \tag{6.18}$$

is the minimum of Problem (6.13) with $\boldsymbol{\theta}$ fixed. It is also defined the function

$$\begin{array}{rl} a_{\sup}: & \mathbb{IR}^{n_\theta} \mapsto \mathbb{IR} \\ & \boldsymbol{\theta} \to a_\theta^*. \end{array} \tag{6.19}$$

We suppose that an inclusion function of $l$ is available, *i.e.* $l$ has an analytic expression. The synthesis problem can be expressed in a general way as the optimization problem (6.20).

$$\begin{cases} \inf\limits_{\boldsymbol{\theta}\in\Theta} & l(\boldsymbol{\theta}) \\ \text{s.t.} & a_{\sup}(\boldsymbol{\theta}) < 0 \end{cases} \tag{6.20}$$

The constraint of Problem (6.20) ensures the sliding condition, and implies the solving of the analysis problem (6.13). Using interval analysis, it is possible to provide an enclosure of $a_{\sup}$ over a box $[\theta]$ [70]. As a consequence, IBBA can be used to solve Problem (6.20) and SIVIA to characterize the set defined by the constraint :

$$\{\boldsymbol{\theta} \in \Theta, a_{\sup}(\boldsymbol{\theta}) < 0\} \tag{6.21}$$

More generally, such a constraint is called a Semi Infinite Constraint (SIC), since it is equivalent to the infinite set of constraint :

$$\begin{array}{rcl} a_{\sup}(\boldsymbol{\theta}) < 0 & \iff & a_\theta^* < 0 \\ & \iff & a_{\boldsymbol{\theta}}(\boldsymbol{\delta}) < 0,\ \forall \boldsymbol{\delta} \in \Delta. \end{array} \tag{6.22}$$

Optimization problems involving SIC are called Semi Infinite Programs (SIP) and can be solved in a global way with different methods [80, 81], and the characterization of the set defined by SICs has been studied in several works [82, 83].

## 6.3 Application case : AUV Ciscrea heave control

In this section, the proposed technique is applied to the control of the heave axis of AUV Ciscrea. As already mentioned, this kind of system suffers from several types of perturbations and uncertainties, which makes the application of robust control tech-

niques essential [84, 85]. However, given its nonlinear dynamics, conventional robust control techniques can only be applied to linearized models around a given operation point.

For this section the modeling of the heave direction developed in section 3.2, will be taken (third row of Eq. 3.11). This can be rewritten as :

$$(M_{RBz} + M_{Az})\ddot{z} + D_{NLz}|\dot{z}|\dot{z} + D_{Lz}\dot{z} + g(z)_z = \tau_{pro_z} + \tau_{envz} \tag{6.23}$$

Here the parameters involved are the coefficients of the heave direction of Eq. 3.11, noted with the subindex $z$. In particular $\tau_{pro_z}$ and $\tau_{envz}$ represent the resultant force in the heave direction produced by the AUV's thrusters and the external perturbations, respectively. It is considered that the coupling with the other directions is negligible.

Let assume that a closed-loop dynamics of the form :

$$\sigma = \dot{e} + \lambda e = 0 \tag{6.24}$$

is desired, with $e = z_d - z$ the tracking error, $z_d$ the position reference and $\lambda$ an approaching rate tuning parameter.

A SM control can provide a solution to this requirement. The structure of a SM controller can be defined with a sliding surface $\sigma = 0$, and implemented by a discontinuous control action of the form :

$$\tau_{pro} = |\tau_{max}| sign(\sigma) \tag{6.25}$$

with $\tau_{max}$ the saturation value of actuator. Once we have the structure defined, it is is necessary to check if the SM establishment conditions are fulfilled, it means to check if condition in Eq.6.6 is fulfilled. The $u_{eq}$ can be obtained as :

$$u_{eq} = -\lambda\dot{z}(M_{RBz} + M_{Az}) + D_{Lz}\dot{z} + D_{NLz}|\dot{z}|\dot{z} + g(z)_z - \tau_{envz} \tag{6.26}$$

For the case of the Ciscrea AUV, referring to Eq. 6.6,

$$|u^-| = |u^+| = \tau_{max} = 6 \quad [\text{Nm}]$$

in the nominal case. It is worth mentioning that Eq. 6.26 depends on the modeling parameters so to guarantee the working conditions of the controller an analysis of their

variations must be considered.

From the SM existence condition, we will deal with the problem of determining which is the fastest dynamics that can be achieved given a domain of the system variables. This implies to solve the problem (6.20) with $l(\boldsymbol{\theta}) = -\lambda$. In this particular case, we can establish the following variable relation :

$$\begin{cases} \boldsymbol{\theta} & \leftrightarrow & \lambda \\ \Theta & \leftrightarrow & [0,2] \\ \boldsymbol{\delta} & \leftrightarrow & (\dot{z}, \tau_{env_z})^T \\ \Delta & \leftrightarrow & ([-0.15, 0.15], [-3,3])^T \end{cases} \tag{6.27}$$

where the ranges of values selected are according to the robot under consideration [49].

Applying the IBBA algorithm, it provides $[-0.3885, -0.3842]$ as an enclosure of the minimum. So the best feasible point found, with respect to the sliding condition, is $\lambda = 0.3842$. In addition it is guaranteed that no value of $\lambda$ greater than $0.3885$ exists such as the sliding condition holds over $\Delta$.

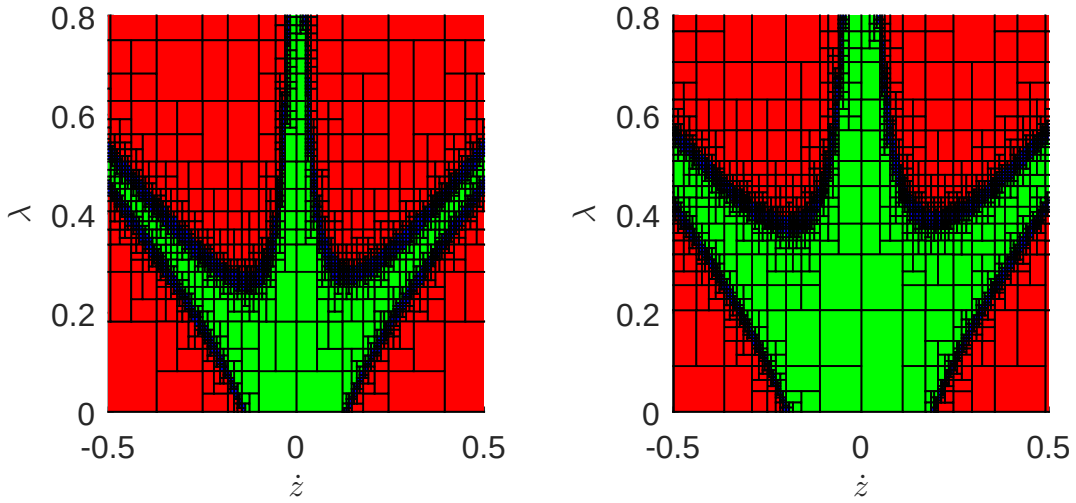
The previous result is the best tuning for the proposed controller, now we are concerned about its limits of applicability with respect to system states and parameters variation. For doing this, we will develop a set of maps to know where the SM is guaranteed.

To begin, we build the subpaving $\lambda$ vs $\dot{z}$. This means to solve the problem established by Eq. 6.21, where the following variable relation is done :

$$\begin{cases} \boldsymbol{\theta} & \leftrightarrow & (\lambda, \dot{z}) \\ \Theta & \leftrightarrow & ([0,2], [-1,1])^T \\ \boldsymbol{\delta} & \leftrightarrow & (\tau_{env_z}) \\ \Delta & \leftrightarrow & ([-3,3])^T \end{cases} \tag{6.28}$$

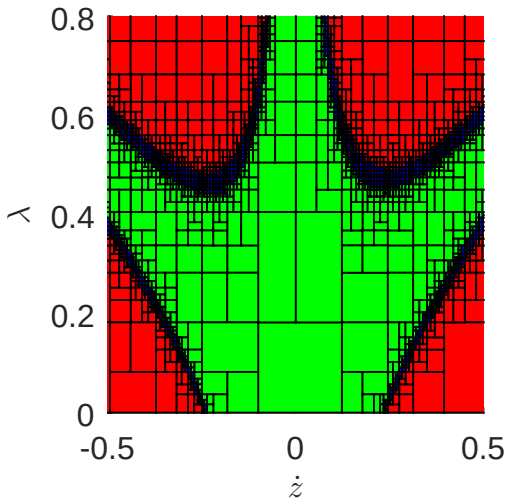
The results are obtained applying the SIVIA algorithm with $\epsilon = 0.01$ and shown in Fig. 6.2 for three values of the total torque produced by the AUV motors. In the subpaving figure red boxes imply no satisfaction of the imposed conditions, green boxes satisfaction of them and blue boxes indicates that the algorithm can not determine the conditions fulfillment. As expected, the observed results show that for low reference speed the dynamics which can be imposed to the system is faster than in those cases where $\dot{z}$ is larger (observe the area around the $\dot{z} = 0$ axis). This has a physical mea-

ning, at higher speeds the inertia of the AUV requires more control action to force the sliding over the desired surface, consequently the possible dynamics are going to be slower. Also Fig. 6.2 shows two lateral “branches" of the feasible area, this behavior obeys to the nonlinear damping term ($D_{NLz}$) present in the AUV model and in the expression of the $u_{eq}$ (Eq.6.26).




(a) $|\tau_{max}| = 4.5$

(b) $|\tau_{max}| = 6$




(c) $|\tau_{max}| = 7.5$

FIGURE 6.2 – SM subpaving analysis for an exact modeling of the system. Green : satisfaction of SM conditions, Red : no satisfaction, Blue : not determined

**Remark 3.** *As we indicated, the red boxes in the subpaving figures imply no satisfaction of the imposed conditions. Note however that this does not mean that SM can not exist, but at least exists one combination of the system parameters where the SM is not going to be established.*

In addition, it has been observed which is the effect of decreasing and increasing the total torque produced by the AUV motors, something that can occur due to battery voltage variations in long operation times. In Fig. 6.2a, it can be appreciated the result of a $25\%$ torque reduction, while in Fig. 6.2c an increment of $25\%$. Both cases are with respect to the Fig. 6.2b where the nominal torque is used. From these figures it is possible to observe how the feasible zone narrows (Fig. 6.2a) or widens (Fig. 6.2c) according to the variation of the maximum control action. From the presented variations we can conclude that even with variations of $25\%$ in the control signal, as our AUV works in the velocity zone $\dot{z} \in [-.1, .1]$ we can impose closed loop dynamics in the interval $\lambda \in (0, 0.27]$.

In Fig. 6.3 simulations of the system step response for three tuning conditions, affected by a constant disturbance $\tau_{env_z} = 3$, are presented. In these simulations the initial position is $z = 3$[m] with the disturbance already established, and at $t = 0$ a reference step of amplitude $-1.5$[m] is applied.

In these simulations the system is tested over three different $\lambda$ values. In all the cases the AUV starts with an initial error, and according to the tuned dynamics its error converges to zero. Additionally, we can observe in Fig. 6.4 the corresponding state trajectories for these simulations.

It is possible to see from the design values ($\lambda$) and the speeds ($\dot{z}$) in these cases, that two of the chosen parameters ($\lambda = 0.2$ and $\lambda = 0.38$) belong to the admissible zone of Fig. 6.2b. So, their time response is product of the sliding behaviour over the designed surface. The third simulation ($\lambda = 0.5$) does not belong to this zone, indeed we see in Fig. 6.4 how the system reaches the sliding surface (dotted line - point D) but SM does not establish, in concordance with the results of Fig. 6.2b, producing an overshoot in the step response (see Fig. 6.3).

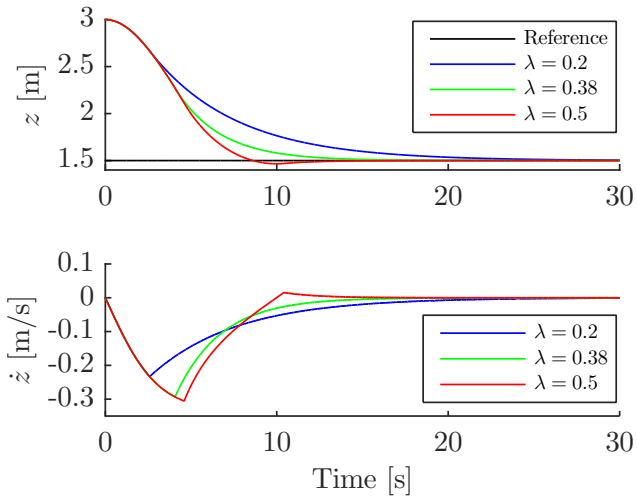



FIGURE 6.3 – System step response for different tuning conditions

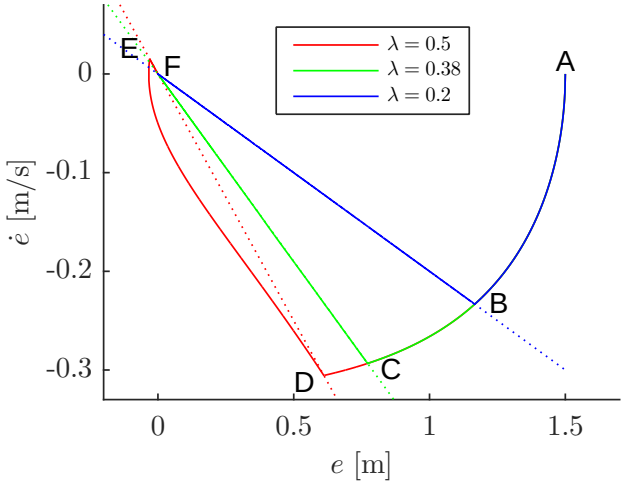


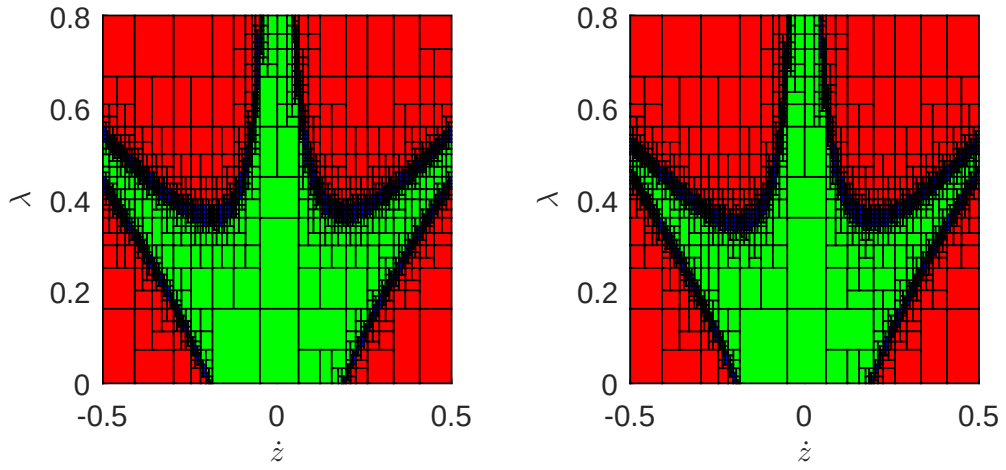

FIGURE 6.4 – Space state evolution for different tuning conditions

Using the same tool, it is possible to analyze how the guaranteed zone of the sliding condition is affected by the parameter uncertainty. For example, Fig. 6.5 shows how variations of $5\%$, $10\%$ and $25\%$ in the $M_{az}$ parameter narrow the guaranteed area of the SM establishment conditions.

For this case referring to Eq. 6.21 the variable relation results in :

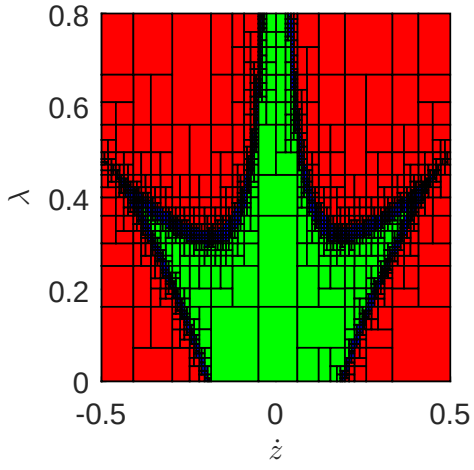
$$\begin{cases} \boldsymbol{\theta} & \leftrightarrow & (\lambda, \dot{z}) \\ \Theta & \leftrightarrow & ([0,2],[-1,1])^T \\ \boldsymbol{\delta} & \leftrightarrow & (\tau_{envz}, M_{Az})^T \\ \Delta & \leftrightarrow & ([-3,3],[63.78,70.49])^T \end{cases} \tag{6.29}$$

where the variance interval of the parameter $M_{Az}$ is adapted for each analysis.




(a) 5% variation of $M_{Az}$

(b) 10% variation of $M_{Az}$




(c) 25% variation of $M_{Az}$

FIGURE 6.5 – SM subpaving analysis considering $M_{Az}$ variation

Furthermore the chosen approach is useful to analyze combined effects of the system uncertainties, Fig. 6.6 shows which will be the effect of a $25\%$ variation simultaneously in the knowledge of $M_{az}$ and $D_{NLz}$. For this case the variable relation results :

$$\begin{cases} \boldsymbol{\theta} & \leftrightarrow & (\lambda, \dot{z}) \\ \Theta & \leftrightarrow & ([0,2],[-0.5,0.5])^T \\ \boldsymbol{\delta} & \leftrightarrow & (\tau_{envz}, M_{Az}, D_{NLz})^T \\ \Delta & \leftrightarrow & ([-3,3],[50.35,83.92],[60.27,100.46])^T \end{cases} \tag{6.30}$$

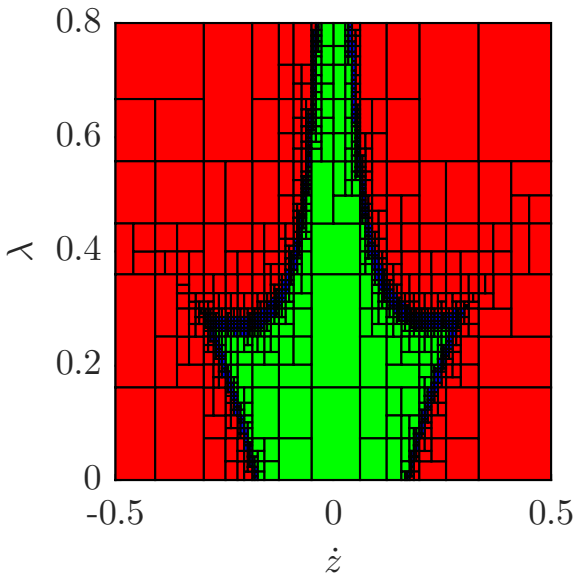



FIGURE 6.6 – SM subpaving analysis considering $M_{Az}$ and $D_{NLz}$ variation

To continue our analysis we can investigate which is the minimal control action we need for a desired $\lambda$ value, this implies to do subpaving graph with $\lambda$ and $|\tau_{max}|$, i.e. to solve Eq. 6.21, where the following variable relation is done :

$$\begin{cases} \boldsymbol{\theta} & \leftrightarrow & (\lambda, |\tau_{max}|)^T \\ \Theta & \leftrightarrow & ([0,2],[-7.5,7.5])^T \\ \boldsymbol{\delta} & \leftrightarrow & (\dot{z}, \tau_{envz})^T \\ \Delta & \leftrightarrow & ([-0.18,0.18],[-3,3])^T \end{cases} \tag{6.31}$$

The result is shown in the Fig.6.7, where it is possible to see that for a control input of $\tau_{max} = 6$ we can reach dynamics up to $\lambda = 0.38$. This is coincident with the result

obtained in the first synthesis problem and in Fig. 6.2b.

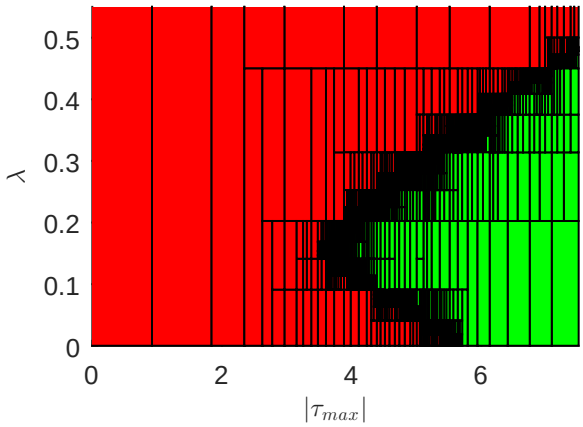



FIGURE 6.7 – Reachable dynamics for $|\dot{z}| \leq 0.18$ and $|\tau_{envz}| = 3$

**Remark 4.** *Although in this application case the study is limited to classic sliding regimes, we highlight the potentiality of the method to other versions of sliding regimes. For example, Fig. 6.7 could serve as a map on how to change the amplitude of control according to the state of the system in an adaptive sliding regime. Also, the same procedure could be follow to design SMRC techniques such as those described in chapters 3 and 4.*

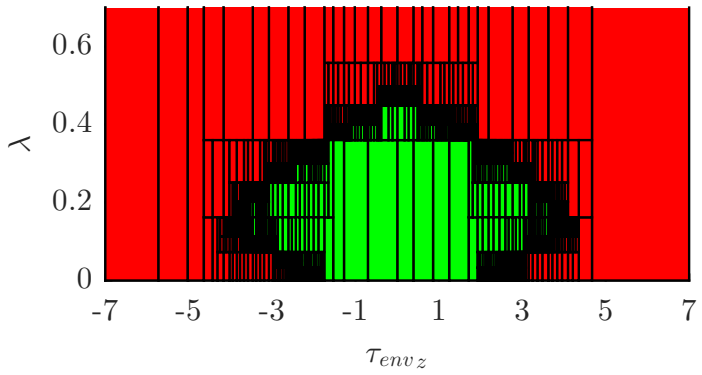
To conclude our analysis it is proposed to see which would be the most extreme perturbation that the system can support guaranteeing the SM conditions. For this, it is generated the subpaving graph with $\lambda$ and $\tau_{envz}$, solving Eq.6.21, with the following variable relation :

$$\begin{cases} \boldsymbol{\theta} & \leftrightarrow & (\lambda, \tau_{envz})^T \\ \Theta & \leftrightarrow & ([0, 0.8], [-7, 7])^T \\ \boldsymbol{\delta} & \leftrightarrow & (\dot{z}) \\ \Delta & \leftrightarrow & ([-0.18, 0.18]) \end{cases} \tag{6.32}$$

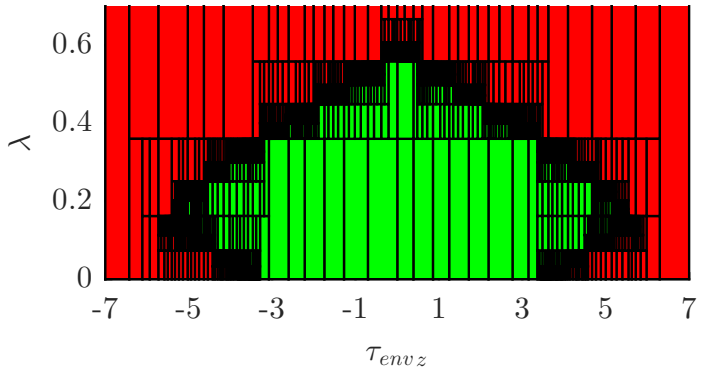
The results are shown in Fig.6.8, where the variation of the applied control signal is also considered.

Note in the nominal case Fig. 6.8b that up to values of $|\tau_{env}| = 3$ the system will work in the guaranteed zone for the domain analyzed. Fig. 6.8a shows how this zone shrinks for a reduction of $25\%$ in $|\tau_{max}|$, and Fig. 6.8c how it is expanded for a $25\%$ increase

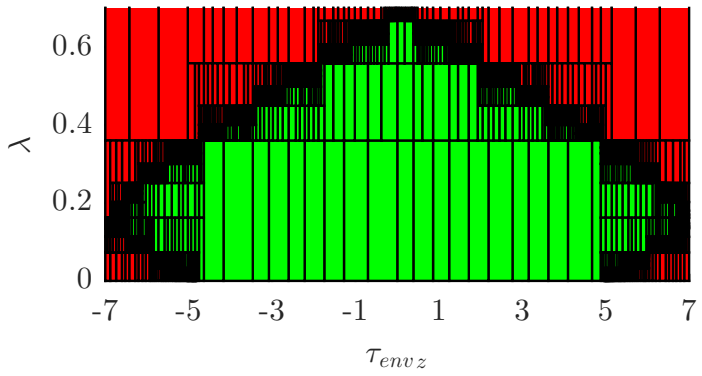
in $|\tau_{max}|$, allowing us to predict the disturbance rejection of the system considering different SM tunings and possible power variations in the AUV.




(a) $|\tau_{max}| = 4.5$




(b) $|\tau_{max}| = 6$




(c) $|\tau_{max}| = 7.5$

FIGURE 6.8 – SM subpaving analysis considering $\tau_{env\,z}$ effects

## 6.4 Conclusions

A technique for the tuning of SM was devolved through the application of global optimization tools. The chosen approach allows us to get a series of maps of the zones where the SM is guaranteed, taking in consideration the possible variation of the system parameters, states values and external perturbations.

Using the interval analysis tools to solve a non-convex global optimization problem, our approach optimizes the SM design for a given criterion. Furthermore, it adds robustness and guarantees the SM set up in front of the process variations and the constrained analyzed state space. To do so, global optimization methods must be used since the synthesis and analysis problems are not convex contrary to the problems emerging in the stochastic approaches which are generally formulated as linear matrix inequalities (therefore convex). However, it must be mentioned that complexity of IBBA grows exponentially with the number of variables, and may fail to solve very large problems.

It is necessary to remark also, that by means of the technique presented in this chapter, the analysis of the sufficient conditions of existence on any kind of sliding regime can be addressed. Particularly the techniques proposed in chapters 3 and 4 can be analyzed with this methodology to obtain guaranteed regions in the state-space where the techniques work.

In the application case, a SM control was designed and analyzed for the heave direction of the AUV Ciscrea, exploring the regions that guarantee SM establishment when considering different operating conditions, disturbances and uncertainties bounds, tuning parameters and control amplitudes. It can be pointed out that not only guarantee areas were obtained, but also estimations of the limit values of parametric uncertainties and perturbations that compromise the SM operation were found.

The work developed in this chapter has led to the presentation of a conference paper [86] (with its publication in IFAC-Papers-OnLine) and a journal article in progress [87].

CHAPITRE 7

# CONCLUDING REMARKS

This work has made a brief review of the principal types of robots, presenting some drawbacks in the navigation and control techniques used in this kind of systems. Throughout the work, some problems of mobile robotic systems have been addressed and posed in terms of constrained control systems. Furthermore, design control techniques capable of dealing with these constrained control problems have been proposed as the main contribution of this work.

With respect to the constraints affecting the input of mobile robotic systems, in chapter 3 advances have been made in mitigating their undesired effects through the application of the SMRC technique. Particularly, SMRC has been applied to path following in the presence of torque constraints. It is important to remark that in this Thesis the SMRC has been extended for its application to the dynamic problem, instead of the kinematic one addressed in previous works. It is left as a future research line the application of this technique in the detection and correction of situations that violate the conditions of kinematic modeling.

Continuing with the idea of using an auxiliary loop to improve the performance achieved by traditional controllers in the presence of constraints, in chapter 4 progress has been made with the tracking problem over restricted paths in presence of mobile obstacles. In this case, the restrictions imposed by the environment were interpreted as restrictions on the possible positions of the robot, therefore as output constraints of the system. In this context, the technique called CASA was developed, demonstrating its efficiency in the obstacle avoidance problem. Although a particular situation has been addressed, it is not discarded that this technique can be applied in contexts of more than one robot, for example, collaborative transport techniques where the constraints will be given by maximum forces to be applied or where virtual constraints could be useful for robot coordination.

The control algorithms developed in chapters 3 and 4 to deal with input and output constraints are implemented as auxiliary control loops without modifying the main

controller of the robotic systems. This is an important advantage of the proposals since, frequently, the controller is embedded in the robotic system.

Another important problem in robotic control is the robustness against model uncertainties and external disturbances. This is particularly true in AUV systems where the hydrodynamic parameters are difficult to estimate and their working environment are also usually unknown. Accordingly, chapter 5 and chapter 6 of this Thesis address the problem of robust design of the robotic control system. The material in these chapters is ordered in terms of the degree of freedom in the control tuning and design. That is, chapter 5 assumes that the main controller is fixed or embedded in the robotic system, but there is freedom in tuning its parameters. On the contrary, chapter 6 assumes that also the structure of the controller can be chosen arbitrarily.

The case treated in chapter 5 can be viewed as the problem of control with structural constraints, i.e. when the structure of the controller is fixed. A PID control structure, the most widespread in mobile robotic applications for tracking, is assumed to be imposed. Then, a methodology to tune this controller guaranteeing performance specifications in the presence of model perturbations is developed.

Finally, chapter 6 addresses the problem of tuning nonlinear robust controllers. A methodology is developed to check, through subpaving generation, if a SM controller tuning achieves the specifications for a given interval of possible perturbations. Furthermore, a method to obtain the optimal SM tuning is also proposed. The presented methodology can be used to tune the SMRC and CASA algorithms developed in chapter 3 and 4. Although the development presented has been based on traditional SM, with little effort it can be extended to other forms of SM as adaptive or higher order ones.

It should not be missed that in chapter 6 the generated subpavings provide useful information on compliance of sufficient conditions for the establishment of SM. In this way, independently of the application to robotics, a problem that is usually solved by considering oversized values to guarantee operation has been addressed here. This allows knowing the necessary control excursion for the particular situation of the system to satisfy the SM conditions.

Globally, this Thesis has meant a contribution to the control of autonomous systems, particularly those affected by restrictions, recognizing the need for robust controls and the synthesis of these in non-linear systems with poor modeling knowledge. It is hoped that this work will serve as a starting point towards more complex problems that can

use the knowledge developed throughout this work. As future works, these techniques could be extended to collaborative mapping and multiple robot coordination applications.

It is important to highlight, as a result of this Thesis, the knowledge development in a strategic area with partial vacancy in the Faculty of Engineering at UNLP. This work also allowed, through research work and mutual collaboration, an international linkage between two research groups at ENSTA Bretagne and UNLP. The development of this work is expected to be the basis for a long-term scientific collaboration as well as a new area of research and application at UNLP.

# BIBLIOGRAPHY


[1] Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, and Giuseppe Oriolo. Robotics. Advanced Textbooks in Control and Signal Processing. Springer-Verlag London, 2009.

[2] Peter Corke. Robotics, Vision and Control : Fundamental Algorithms in MATLAB. Springer Tracts in Advanced Robotics 73. 2011.

[3] Thor Fossen. Marine control systems : guidance, navigation and control of ships, rigs and underwater vehicles, volume 28. December 2002.

[4] Oussama Khatib Bruno Siciliano, editor. Springer Handbook of Robotics. Springer-Verlag Berlin Heidelberg, 1st edition, 2008.

[5] Rached Dhaouadi and Ahmad Abu Hatab. Dynamic modelling of differential-drive mobile robots using lagrange and newton-euler methodologies : A unified framework. 2013.

[6] Luc Jaulin. Automation for Robotics. Automation for Robotics, 2015.

[7] Shimon Y. Nof, editor. Handbook of Industrial Robotics. Wiley, 2nd edition, March 1999.

[8] Seth Hutchinson George Kantor Wolfram Burgard Lydia E. Kavraki Sebastian Thrun Howie Choset, Kevin M. Lynch. Principles of Robot Motion : Theory, Algorithms, and Implementations. Intelligent Robotics and Autonomous Agents. The MIT Press, 2005.

[9] Daniel Sack and Wolfram Burgard. A comparison of methods for line extraction from range data. IFAC Proceedings Volumes, 37(8) :728 – 733, 2004. IFAC/EURON Symposium on Intelligent Autonomous Vehicles, Lisbon, Portugal, 5-7 July 2004.

[10] Michael Montemerlo, Sebastian Thrun, Daphne Koller, and Ben Wegbreit. Fastslam : A factored solution to the simultaneous localization and mapping problem. In In Proceedings of the AAAI National Conference on Artificial Intelligence, pages 593–598. AAAI, 2002.

[11] Patrick Pfaff, Rudolph Triebel, and Wolfram Burgard. An efficient extension to elevation maps for outdoor terrain mapping and loop closing. The International Journal of Robotics Research, 26(2) :217–230, 2007.

[12] Nathaniel Fairfield, George Kantor, and David Wettergreen. Real-time slam with octree evidence grids for exploration in underwater tunnels. Journal of Field Robotics, 24 :3–21, 2007.

[13] Hongyan Guo, Dongpu Cao, Hong Chen, Zhenping Sun, and Yunfeng Hu. Model predictive path following control for autonomous cars considering a measurable disturbance : Implementation, testing, and verification. Mechanical Systems and Signal Processing, 118 :41 – 60, 2019.

[14] L. Lapierre, D. Soetanto, and A. Pascoal. Nonlinear path following with applications to the control of autonomous underwater vehicles. In 42nd IEEE International Conference on Decision and Control (IEEE Cat. No.03CH37475), volume 2, pages 1256–1261, 2003.

[15] X. Xiang, C. Yu, and Q. Zhang. Robust fuzzy 3d path following for autonomous underwater vehicle subject to uncertainties. Computers and Operations Research, 84 :165–177, 2017. cited By 88.

[16] F. Garelli, L. Gracia, A. Sala, and P. Albertos. Switching algorithm for fast robotic tracking under joint speed constraints. In Control & Automation (MED), 2010 18th Mediterranean Conference on, pages 802–807, 23-2.

[17] N. Nenchev Dragomir and Uchiyama Masaru. Singularity-consistent path planning and motion control through instantaneous self-motion singularities of parallel-link manipulators. J. Robotic Syst., 14(1) :27–36, July 1997.

[18] Lionel Lapierre, Rene Zapata, and Pascal Lepinay. Combined path-following and obstacle avoidance control of a wheeled robot. The International Journal of Robotics Research, 26(4) :361–375, 2007.

[19] Siciliano Bruno Bastin Georges Canudas de Wit, Carlos, editor. Theory of Robot Control. Springer, London, 1996.

[20] Chae H. An, Christopher G. Atkeson, and John M. Hollerbach. Model-based Control of a Robot Manipulator. MIT Press, Cambridge, MA, USA, 1988.

[21] J. L. Crowley. World modeling and position estimation for a mobile robot using ultrasonic ranging. In Proceedings, 1989 International Conference on Robotics and Automation, pages 674–680 vol.2, 14-1.

[22] P. Moghadam, W. S. Wijesoma, and Dong Jun Feng. Improving path planning and mapping based on stereo vision and lidar. In 2008 10th International Conference on Control, Automation, Robotics and Vision, pages 384–389, 17-2.

[23] Y. Koren and J. Borenstein. Potential field methods and their inherent limitations for mobile robot navigation. In Proceedings. 1991 IEEE International Conference on Robotics and Automation, pages 1398–1404 vol.2, 9-11.

[24] Yi Zhu, Tao Zhang, Jingyan Song, Xiaqin Li, and Masatoshi Nakamura. A new method for mobile robots to avoid collision with moving obstacle. Artificial Life and Robotics, 16(4) :507–510, February 2012.

[25] D. Wilkie, J. van den Berg, and D. Manocha. Generalized velocity obstacles. In 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 5573–5578, 10-1.

[26] C. Kunz, C. Murphy, R. Camilli, H. Singh, J. Bailey, R. Eustice, M. Jakuba, K. i. Nakamura, C. Roman, T. Sato, R. A. Sohn, and C. Willis. Deep sea underwater robotic exploration in the ice-covered arctic ocean with auvs. In 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 3654–3660, 22-2.

[27] Russell B. Wynn, Veerle A. I. Huvenne, Timothy P. Le Bas, Bramley J. Murton, Douglas P. Connelly, Brian J. Bett, Henry A. Ruhl, Kirsty J. Morris, Jeffrey Peakall, Daniel R. Parsons, Esther J. Sumner, Stephen E. Darby, Robert M. Dorrell, and James E. Hunt. Autonomous underwater vehicles (auvs) : Their past, present and future contributions to the advancement of marine geoscience. Marine Geology, 352 :451–468, June 2014.

[28] M. Dunbabin and L. Marques. Robots for environmental monitoring : Significant advancements and applications. IEEE Robotics & Automation Magazine, 19(1) :24–39, February 2012.

[29] R. L. Wernli. Auv commercialization-who's leading the pack ? In OCEANS 2000 MTS/IEEE Conference and Exhibition. Conference Proceedings (Cat. No.00CH37158), volume 1, pages 391–395 vol.1, 2000.

[30] Dennis S. Bernstein and Anthony N. Michel. A chronological bibliography on saturating actuators. International Journal of Robust and Nonlinear Control, 5 :375–380, August 1995.

[31] E. Campos, A. Chemori, V. Creuze, J. Torres, and R. Lozano. Saturation based nonlinear depth and yaw control of underwater vehicles with stability analysis and real-time experiments. Mechatronics, 45 :49–59, August 2017.

[32] Zewei Zheng, Cheng Jin, Ming Zhu, and Kangwen Sun. Trajectory tracking control for a marine surface vessel with asymmetric saturation actuators. Robotics and Autonomous Systems, 97 :83–91, November 2017.

[33] Leo V. Steenson, Alexander B. Phillips, Eric Rogers, Maaten E. Furlong, and Stephen R. Turnock. Experimental verification of a depth controller using model predictive control with constraints onboard a thruster actuated auv. IFAC Proceedings Volumes, 45(5) :275–280, January 2012.

[34] Pouria Sarhadi, A. Ranjbar, and Alireza Khosravi. Adaptive integral feedback controller for pitch and yaw channels of an auv with actuator saturations. ISA Transactions, 65, September 2016.

[35] Y. Wang, L. Gu, Y. Xu, and X. Cao. Practical tracking control of robot manipulators with continuous fractional-order nonsingular terminal sliding mode. IEEE Transactions on Industrial Electronics, 63(10) :6194–6204, October 2016.

[36] Yaoyao Wang, Fei Yan, Surong Jiang, and Bai Chen. Time delay control of cable-driven manipulators with adaptive fractional-order nonsingular terminal sliding mode. Advances in Engineering Software, 121 :13–25, July 2018.

[37] Yaoyao Wang, Surong Jiang, Bai Chen, and Hongtao Wu. A new continuous fractional-order nonsingular terminal sliding mode control for cable-driven manipulators. Advances in Engineering Software, 119 :21–29, May 2018.

[38] Dragomir N. Nenchev. Tracking manipulator trajectories with ordinary singularities : A null space-based approach. International Journal of Robotics Research, 14(4) :399–404, 1995.

[39] Fabricio Garelli, Luis Gracia, Antonio Sala, and Pedro Albertos. Sliding mode speed auto-regulation technique for robotic tracking. Robotics and Autonomous Systems, 59(7) :519–529, July 2011.

[40] R. Yang, B. Clement, A. Mansour, H. J. Li, and M. Li. Robust heading control and its application to ciscrea underwater vehicle. In OCEANS 2015 - Genova, pages 1–6, 18-2.

[41] Gianluca Antonelli. Underwater Robots : Motion and Force Control of Vehicle-Manipulator Systems. Springer Tracts in Advanced Robotics. Springer, 2006.

[42] Rui Yang, Benoit Clement, Ali Mansour, Ming Li, and Nailong Wu. Modeling of a complex-shaped underwater vehicle for robust control scheme. Journal of Intelligent & Robotic Systems, 80(3) :491–506, December 2015.

[43] R. Yang, B. Clement, A. Mansour, H. J. Li, M. Li, and N. L. Wu. Modeling of a complex-shaped underwater vehicle. In 2014 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), pages 36–41, 14-1.

[44] Vadim Utkin, Juergen Guldner, and Jingxin Shi. Sliding Mode Control in Electro-Mechanical Systems. January 2009.

[45] Fabricio Garelli, Ricardo J. Mantz, and Hernán De Battista. Advanced control for constrained processes and systems. Institution of Engineering and Technology, 2011.

[46] Juan Luis Rosendo, Benoit Clement, and Fabricio Garelli. Sliding mode reference conditioning for path following applied to an AUV. IFAC-PapersOnLine, 49(23) :8–13, January 2016. CAMS.

[47] Juan Luis Rosendo, Fabricio Garelli, Benoit Clement, and Hernán De Battista. Mitigation of the saturation effect in AUV path following applications. In AADECA 2016 - Semana del Control Automático - 25º Congreso Argentino de Control Automático. Asociación Argentina de Control Automático (AADECA)., Buenos Aires, 2016. ISBN 978-950-99994-9-7.

[48] Juan Luis Rosendo, Benoit Clement, and Fabricio Garelli. Acondicionamiento de la referencia utilizando modos deslizantes en aplicaciones de seguimiento de camino en AUV. In Cuartas Jornadas de Investigación, Transferencia y Extensión de la Facultad de Ingeniería., La Plata, Argentina, 2017. UNLP. Facultad de Ingeniería, Universidad Nacional de La Plata.

[49] Juan Luis Rosendo, Benoit Clement, and Fabricio Garelli. Experimental validation of constraint mitigation algorithm in underwater robot depth control. Proceedings of the Institution of Mechanical Engineers, Part I : Journal of Systems and Control Engineering, 233(3) :264–275, August 2018.

[50] Y. Zhao and P. Ioannou. Positive train control with dynamic headway based on an active communication system. IEEE Transactions on Intelligent Transportation Systems, 16(6) :3095–3103, December 2015.

[51] E. J. de Oliveira L. Oliveira da Costa, M. B. Arantes de Souza and L. Willer de Oliveira. Optimal speed of multiple UAVs based on nonlinear programming to avoid collision in periodic paths. In EngOpt 2016 - 5th International Conference on Engineering Optimization, 2016.

[52] Siciliano B., Sciavicco L., Villani L., and Oriolo G. Robotics. Modelling, Planning and Control. Springer, London, 2009.

[53] E. Rohmer, S. P. N. Singh, and M. Freese. V-rep : A versatile and scalable robot simulation framework. In 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 1321–1326, 3-7.

[54] Juan Luis Rosendo, Fabricio Garelli, Hernan De Battista, and Fernando Valenciaga. Obstacle avoidance under strict path following. In XVII Workshop on Information Processing and Control (RPIC)., Argentina, Mar del Plata, 2017. IEEE.

[55] Juan Luis Rosendo, Fabricio Garelli, and Hernán De Battista. Obstacle avoidance with path restrictions in autonomous underwater vehicles. In AADECA 2018 - Semana del Control Automático - 26º Congreso Argentino de Control Automático. Asociación Argentina de Control Automático (AADECA)., 2018.

[56] J. L. Rosendo, H. De Battista, and F. Garelli. Obstacle avoidance with path restrictions in autonomous underwater vehicles. Sent to Advances in Mechanical Engineering, 2018.

[57] Zhengping Feng and Robert Allen. Reduced order $h\infty$ control of an autonomous underwater vehicle. IFAC Proceedings Volumes, 36(4) :121–126, April 2003.

[58] Jan Petrich and Daniel J. Stilwell. Robust control for an autonomous underwater vehicle that suppresses pitch and yaw coupling. Ocean Engineering, 38(1) :197–204, January 2011.

[59] Kiam Heong Ang, G. Chong, and Yun Li. Pid control system analysis, design, and technology. IEEE Transactions on Control Systems Technology, 13(4) :559–576, 2005.

[60] A. O'Dwyer. Handbook of PI and PID Controller Tuning Rules. Imperial College Press, 2009.

[61] Ming Ge, Min-Sen Chiu, and Qing-Guo Wang. Robust pid controller design via lmi approach. Journal of Process Control, 12(1) :3–13, January 2002.

[62] K. Li. Pid tuning for optimal closed-loop performance with specified gain and phase margins. IEEE Transactions on Control Systems Technology, 21(3) :1024–1030, 2013.

[63] Linlin Ou, Weidong Zhang, and Danying Gu. Nominal and robust stability regions of optimization-based pid controllers. ISA Transactions, 45(3) :361–371, July 2006.

[64] Karl J. Aström and Tore Hägglund. PID Controllers : Theory, Design, and Tuning. International Society of Automation, second edition edition, 1995.

[65] Fung Ho-Wang, Wang Qing-Gup, and Lee. Tong-Heng. Pi tuning in terms of gain and phase margins. Automatica, 34(9) :1145–1149, September 1998.

[66] Ye. Smagina and Irina Brewer. Using interval arithmetic for robust state feedback design. Systems & Control Letters, 46(3) :187–194, July 2002.

[67] S. Khadraoui, M. Rakotondrabe, and P. Lutz. Pid-structured controller design for interval systems : Application to piezoelectric microactuators. In Proceedings of the 2011 American Control Conference, pages 3477–3482, 2011.

[68] P. Apkarian, M. N. Dao, and D. Noll. Parametric robust structured control design. IEEE Transactions on Automatic Control, 60(7) :1857–1869, 2015.

[69] R. Baker Kearfott. An interval branch and bound algorithm for bound constrained optimization problems. Journal of Global Optimization, 2(3) :259–280, September 1992.

[70] D. Monnet, J. Ninin, and B. Clément. A global optimization approach to structured regulation design under $H_\infty$ constraints. 55th IEEE Conference on Decision and Control (CDC), Las Vegas, 2016.

[71] Kemin Zhou and John Comstock Doyle. Essentials of robust control, volume 104. Prentice hall Upper Saddle River, NJ, 1998.

[72] Pierre Apkarian and Dominikus Noll. Nonsmooth h$infty$-synthesis. IEEE Trans. Automat. Contr., 51 :71–86, January 2006.

[73] James Burke, D. Henrion, Adrian Lewis, and Michael Overton. Hifoo-a matlab package for fixed-order controller design and $h_\infty$ optimization. IFAC Proceedings Volumes, 39(9) :339–344, April 2006.

[74] Olivier Didrit Eric Walter Luc Jaulin, Michel Kieffer. Applied Interval Analysis. Springer-Verlag London, 1st edition, 2001.

[75] P. M. Young, M. P. Newlin, and J. C. Doyle. Mu analysis with real parametric uncertainty. In Proceedings of the 30th IEEE Conference on Decision and Control, volume 2, pages 1251–1256. IEEE, 1991.

[76] Juan Luis Rosendo, Dominique Monnet, Benoît Clement, and Fabricio Garelli. Control of an autonomous underwater vehicule under robustness constraints. In SWIM 2016 (Summer Workshop on Interval Methods)., Lyon, France., 2016. École Normale Supérieure de Lyon (ENS de Lyon).

[77] Juan Luis Rosendo, Dominique Monnet, Benoit Clement, Fabricio Garelli, and Jordan Ninin. Control of an autonomous underwater vehicle subject to robustness constraints. IFAC-PapersOnLine, 51(25) :322–327, January 2018.

[78] J. Ninin. Global Optimization based on Interval Analysis : Affine Relaxation and Limited Memory. PhD thesis, Institut National Polytechnique de Toulouse - INPT, 2010.

[79] R. Moore, R. Kearfott, and M. Cloud. Introduction to Interval Analysis. Society for Industrial and Applied Mathematics, July 2009.

[80] Alexander Mitsos. Global optimization of semi-infinite programs via restriction of the right-hand side. Optimization, 60(10-11) :1291–1308, October 2011.

[81] B. Bhattacharjee, P. Lemonidis, W. H. Green Jr., and P. I. Barton. Global solution of semi-infinite programs. Mathematical Programming, 103(2) :283–307, June 2005.

[82] Alexandre Goldsztejn, Claude Michel, and Michel Rueher. Efficient handling of universally quantified inequalities. Constraints, 14(1) :117–135, 2009.

[83] Stefan Ratschan. Approximate quantified constraint solving by cylindrical box decomposition. Reliable Computing, 8(1) :21–42, February 2002.

[84] Lionel Lapierre. Robust diving control of an AUV. Ocean Engineering, 36(1) :92 – 104, 2009. Autonomous Underwater Vehicles.

[85] Taha Elmokadem, Mohamed Zribi, and Kamal Youcef-Toumi. Trajectory tracking sliding mode control of underactuated AUVs. Nonlinear Dynamics, 84(2) :1079–1091, Apr 2016.

[86] Dominique Monnet, Juan Luis Rosendo, Hernán De Battista, Benoit Clement, Jordan Ninin, and Fabricio Garelli. A global optimization approach for non-linear sliding mode control analysis and design. IFAC-PapersOnLine, 51(25) :128–133, January 2018.

[87] J. L. Rosendo, D. Monnet, H. De Battista, J. Ninin, B. Clement, and F. Garelli. Sliding mode control analysis and design for an AUV application using global optimization techniques. Sent to An International Journal of Nonlinear Dynamics and Chaos in Engineering Systems, 2018.

UNIVERSITE BRETAGNE LOIRE MATHSTIC

ENSTA Bretagne

## Titre : Techniques robustes pour le contrôle automatique des systèmes robotiques

**Mot clés :** Robotique, Contrôle robuste, Systèmes avec restrictions, Mode de glissement

**Resumé :** Ce travail vise à atténuer les effets des contraintes sur les systèmes robotiques mobiles. À cette fin, des structures de commande auxiliaire et des techniques de réglage robuste sont proposées. Les structures sont proposées dans le cadre du suivi de chemin pour atténuer les effets des contraintes sur les entrées et les sorties des systèmes. Ensuite, etant donnée leur utilisation répandue en robotique, les contrôleurs de type PID sont considérés comme une contrainte structurelle. Une méthode de réglage robuste, tenant compte de cette contrainte, est proposée permettant d'atteindre de bons niveaux de performance même en présence de perturbations. Enfin, pour faire face à la robustesse en présence des contraintes de non-linéarité sur robots, un outil d'analyse et de réglage pour les contrôleurs de mode glissant est proposé. La particularité de cette méthode de réglage, basée sur des techniques d'optimisation globale et de calculs par intervalles, est qu'elle permet de générer des cartographies de réglage des paramètres pour lesquels le critère de performance souhaité est rempli. Toutes les stratégies proposées sont mises en pratique par des expérimentations réelles ou sur des simulateurs validés (AUV Ciscrea disponible à l'ENSTA Bretagne).

## Title : Robust techniques of automatic control for mobile robotic systems

**Keywords :** Robotics, Robust control, Systems with constraints, Sliding Modes

**Abstract :** This work seeks to mitigate the effects of constraints on mobile robotic systems. To this end, auxiliary control loops and robust tuning techniques are proposed. The former are proposed to mitigate the effects of constraints on the input and output of the systems through the modification of the motion parameter in path following applications. Then, PID controllers are considered as a structural constraint, given its wide use in robotics particularly at low control level. A robust tuning methodology considering this constraint is proposed which achieves good performance levels even when facing disturbances. Finally, to deal with robustness in presence of robots nonlinearity constraints, an analysis and tuning tool for sliding mode controllers is proposed. The particularity of this tuning method, based on global optimization and interval techniques, is that it allows generating tuning maps of the parameter regions where the desired performance criterion is fulfilled. All the proposed strategies are put into practice, through real experimentation or in validated simulators, over the AUV Ciscrea available at ENSTA Bretagne.