FROM OVERACTUATED TO UNDERACTUATED: PATH-FOLLOWING ALGORITHM FOR ROBOTICS SYSTEMS

14 August 2022

Manas UTESHEV X19

ÉCOLE POLYTECHNIQUE

IP PARIS

l'Europe s'engage en Occitanie
UNION EUROPEENNE
La Région Occitanie Pyrénées - Méditerranée
Projet LEZ 2020 Cofinancé par le fonds Européen au Développement Régional

Contents

1 Abstract

During the last years, the autonomous vehicles have gained a considerable interest among researchers. For example, such robots as self-driving cars, self-flying planes, etc., would pave the way for automation of our day-to-day activities and would assist us in accomplishing tasks, which humans are not able to complete on their own and thus, such workforce will become necessary if we want our societies to progress.

Moreover, in order to resolve problems related to the sustainable development, which is vital to the survival of our society in this fast-changing environment, some scientific researchers have decided to manage projects, which are focused on the exploitation of autonomous vehicles so that they could keep in check the climate change and complete the complex tasks, which either cannot be completed by humans, or require significant costs in terms of finance and human effort, if autonomous vehicles do not engage in these tasks.

Furthermore, my supervisor Lionel Lapierre decided to start a project two years ago, which is supported financially by *European Regional Development Fund* and is called *LEZ 2020*. The aim of this project is to develop the *CUBE* robot, which will build autonomously a map of the LEZ source, that supplies the whole city of Montpellier in the south of France with drinking water. This project is going to bring significant changes to the karst exploration in the future in case of its success. Henceforth, the objective of my research has been to bring solutions, which will accelerate the development of the *CUBE* robot.

To be more precise, my elaborated solutions cover such topics as the passage from overactutated to underactuated robotics systems, the saturation and reactivity of the robot's actuators and the selection criteria for motion control.

(a) *CUBE* robot

(b) LEZ source

Figure 1: Karst exploration

2 Introduction

2.1 Context of the research

Since the industrial revolution in Western Europe the automation of tasks has become an irreplaceable part of our society. If it had not been for an automated process, the industrial progress, from which our society has been profiting considerably, would not even have existed nowadays. It is useful to note that the first automated industrial process was developed by *Oliver Evans* in 1785[1] and in fact, it was an automated floor mill. Since then, in order to prosper in the fast-changing world the industrialists has been trying to implement automation so that they could reduce the costs by decreasing the human labour at their disposal and accelerate the production of goods, which has been vital for a rising demand.

Moreover, we are all aware that the climate change has become a crucial issue for the modern world. Unfortunately, it has raised some challenges, that are going to make us change our everyday habits and to motivate us to find innovative solutions to these problems. That is where the use of autonomous vehicles has become quite important, as it is clear that by using only the human labour we are going to accomplish almost nothing.

Furthermore, the autonomous vehicles are going to help us to study the real pace at which our climate changes and how it actually changes. For example, as we all know the drinking water is important for our everyday lives and its shortage has become a threatening issue to our human existence. If we take a look at the city of Montpellier, located in the south of France near the Mediterranean Sea, then we can see that the drinking water provided to the whole city is actually extracted from underground drainage *karst* systems by the assistance of underground pumping systems. To be more precise, according to the certain source[2], the *karst* is the land feature (in other words, the land form), which is formed as a result of dissolution of the soluble rocks such as *limestone*, *dolomite* and *gypsum*. Usually in order to visualize the *karst* structure, we can refer ourselves to the underground caves. That is why we can say that the city of Montpellier relies heavily in terms of drinking water on the *karst* structure, e.g. underground caves, near the source of *Lez*.

In addition, in order to supply easily the city of Montpellier with drinking water, pumping systems have been installed in the underground caves near the source of *Lez*. However, the climate change has produced a considerable impact on the depth, at which the drinking water flows in underground caves, and thus, because of the increase of such depth the city has decided to install another pumping system at a deeper level in order to satisfy its need in terms of drinking water. Unfortunately, as the climate change is not going to stop in the near future, the level of the underground drinking water is predicted to go deeper in the karst structure, which will in turn require another pumping system due to the lack of alternative. Actually such ground-

[1] https://en.wikipedia.org/wiki/Automation
[2] https://en.wikipedia.org/wiki/Karst

water resource is called *Karst aquifer*[3] and is a productive source of drinking water. However, karst aquifers are vulnerable to the contamination, which is why the supervision of the water's quality in karst aquifers must be put in place as this source provides with drinking water the whole city of Montpellier.

That is why the *Karst exploration* has become the main challenge of the project *LEZ 2020*, on which I have been concentrated during my research internship. Moreover, the principal objective of the project has been to develop the Underwater Autonomous Vehicle (UAV), called *CUBE*, which with the assistance of the motion control, conceived by the team of researchers, of its eight motors is going to navigate in the underground caves on its own and draw a map of the whole karst structure near the source of *Lez* by exploiting sensors installed in this robotic system.

Last but not least, another objective for the robotic system *CUBE* will be to take a sample of drinking water and analyze its quality during its passage through the karst aquifers. Even though the karst aquifers are the productive source of the drinking water, they are vulnerable to the contamination, which can in turn lead to the disastrous consequences for those, who consume such water. That is why the quality of the drinking water is the critical issue for the city of Montpellier, which relies heavily on karst aquifers in terms of drinking water supply. To further strengthen the point of the last objective for the *CUBE* robot, the geologists have analyzed during the years the quality of the drinking water in the karst aquifers near the source of *Lez* and have stumbled upon the chemical elements such as plastic present in the drinking water. In spite of the fact that such plastic is present in the drinking water of the karst aquifers in negligible amounts and thus, such drinking water can be consumed, the level of plastic in the drinking water is predicted to rise due to the pollution of the environment and if the pollution continues at its usual pace and the level of plastic increases considerably, the French authorities of the city of Montpellier will be obliged to install filters near the source of *Lez* in order to provide the whole city with the water of standard quality. That is why the mapping of the karst aquifers and the analysis of the quality of the drinking water are two pillars of the research project *Lez 2020*. The success of this project will provide the French authorities with different sorts of valuable information such as the potential of the karst aquifers in terms of the drinking water supply, the evolution of the level of the water in the underground caves and the evolution of the quality of the drinking water.

2.2 Summary

In this report we are going to analyze the research internship, that I have carried out during 4 months in *Laboratoire d'Informatique, de Robotique et de Microéléctronque de Montpellier*[4] (LIRMM) in Montpellier, France, under the supervision of the LIRMM's assisstant professor *Lionel Lapierre*. LIRMM is currently affiliated with the *University of Montpellier* and *CNRS* and is focused on the research in Computer Science, Robotics and Microelectronics. I have

[3] `https://www.usgs.gov/mission-areas/water-resources/science/karst-aquifers`
[4] `https://www.lirmm.fr/lirmm-en/`

conducted my research internship in the Robotics department of LIRMM, which is at the moment composed of four teams: *DEXTRER*, *ICAR*, *EXPLORE* and *IDH*. During my internship I have been a member of the team *EXPLORE*[5], which is specialized in the research on *Mobile Robotics for environment exploration*. The team is focused mainly on the study of the *Underwater Autononomous Vehicles* (UAVs) and thus, wants to conceive autonomous robotics systems, that are able to fulfill complex tasks on their own given the harsh natural constraints imposed by the aquatic environment and the hardware architecture. In addition, in order to get acquainted with the practical results, which have been obtained by the team, you can visualize them on their *Youtube* channel[6].

At the beginning of my research internship I have been presented a notion, which constitutes a backbone of the research core of the team *EXPLORE* and is called the *Path-following Algorithm*. Moreover, according to the paper [**1**] during the last years the motion control of the autonomous vehicles has been a huge area of interest for researchers and many improvements have been made in this area. In order to clearly see what solutions have been proposed for this area in literature, we can divide all these solutions approximately into three groups as it has been done in the article [**1**]:

- Point stabilization
- Trajectory tracking
- Path following

This report will focus only on the first solution, as it is the most recent among all solutions and is the most promising for the robotics researchers at LIRMM.

After that, I have been studying overactuated and underactuated robotics systems and trying to propose a solution for motion control, when the overactuated system becomes underactuated because of the loss of motors due to environment conditions. As a result, I have worked out a solution, which helps us to point out explicitly the directions, in which our robot can exert a resulting force. After having elaborated my proper solution, I focused on the *selection criteria* for the motion control of UAVs, e.g. the criteria, which determines the feasible directions, in which the energy consumption will be minimal for the robot.

Moreover, I have decided to analyze the problem of the saturation and reactivity constraints, which are imposed by the hardware architecture, and to try to propose a solution, which will satisfy these constraints and minimize the energy consumption at the same time.

[5] `https://www.lirmm.fr/teams-en/EXPLORE-en/`
[6] `https://www.youtube.com/channel/UCeKFkZD3DjEZb1ZSRlxN3Kg`

3 Preliminaries

3.1 Presentation of the methods of the motion control

As it has been mentioned above, the solutions proposed in the area of the motion control for autonomous vehicles can be divided approximately into three groups: *point stabilization*, *trajectory tracking* and *path-following*.

The first solution, which is *point stabilization*, refers to the method, where given the orientation the main objective is to stabilize the autonomous vehicle at a given point. According to the paper [**1**] it becomes quite challenging to use the *point stabilization* method when we have nonintegrable constraints and the continuous state-feedback law does not exist, which in turn does not guarantee stability. In order to resolve such problem, various solutions exploiting hybrid feedback laws and smooth time-varying control laws have been presented.

Secondly, the *trajectory tracking* for the motion control refers to the method, where the autonomous vehicle is given a task to track a reference, which is parameterized by the time. According to the paper [**1**] the *trajectory tracking* method has shown pretty good results when being applied in overactuated robotic systems, e.g. systems where the vehicle has more actuators than state variables. However, the case of underactuated systems, e.g. systems where the robot has less actuators than state variables, is still a challenging topic for the researchers specialized in *trajectory tracking*. The paper [**1**] states that linearization and feedback linearization methods have been proposed in order to resolve such challenge.

Finally, the most promising method for the motion control, which has been introduced quite recently and received less attention than the first two methods, is the *path-following*. It refers to the method, where the autonomous vehicle is given a task to converge to and follow the given path without any temporal restrictions. One of the first promising articles in this area was the paper [**2**] released in 1991 by Samson and Ait-Abderrahim, which will be discussed more in the sections below. Moreover, the application of the *path-following* method on marine autonomous vehicles has been developed by Encarnaçao et al. in the article [**3**]. Encarnaçao et al. have elaborated with the assistance of the *Lyapunov theory* the kinematic controller for the marine autonomous vehicles by assuming that the velocity of the ocean current is known to the robot. After that, in order to analyze the case when the ocean current is unknown the authors of the article [**3**] exploit estimators of the ocean current and modify in turn the kinematic controller. Finally, by backstepping the kinematics into dynamics they introduce the dynamic controller for the marine robot and perform the same analysis on the availability of information about the ocean current to the vehicle.

However, the article of Encarnaçao et al. has only presented results in *2D* space and the expansion to the *3D* space is of course much needed. Moreover, the paper does not take into account the saturation of the actuators, which is always the major problem during the development of the autonomous vehicles and thus, this issue must be resolved in order for the robot to be put in practice and to bring satisfactory results when trying to complete complex tasks in the real-world conditions.

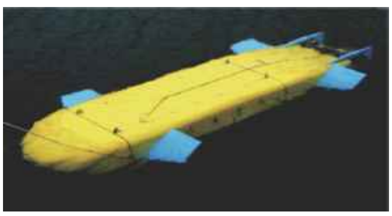
3.2 Path-following control design

The following results and notations are based on the paper [**4**] of Lapierre and Soetanto. The article takes into account the results obtained in the papers [**3**] of Encarnaçao et al. and [**5**] of Micaelli and Samson.

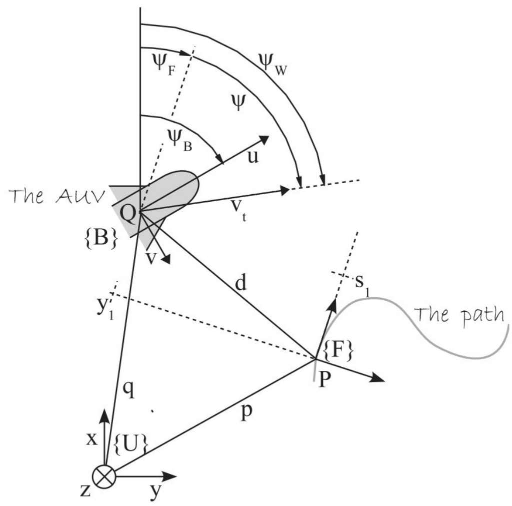
For the ease of visualization let's place ourselves in the *2D* space and let's consider the robot *INFANTE AUV* (autonomous underwater vehicle) developed by the Institute for Systems and Robotics (ISR) of Lisbon, Portugal. Actually, this AUV disposes of two identical symmetric back thrusters. As you can see, this robotic system is clearly underactuated since the lateral thruster is absent in *INFANTE AUV*. You can refer to the figure 2a in order to visualize how *INFANTE AUV* looks like. Moreover, the thrusters generate a force denoted F along the longitudinal axis and a torque denoted Γ around its vertical axis.

3.2.1 Notation of the vehicle's model. Kinematics and dynamics

Even though we consider the autonomous vehicle to be functioning in *3D* space, for the sake of simplicity we suppose that the robot moves in the plane, e.g. the $\mathbf{z}$-component of the coordinates of the robot's position is always zero. In the sequel the symbol $\{A\} := \{\mathbf{x}_A, \mathbf{y}_A, \mathbf{z}_A\}$ will denote the reference frame with respect to the origin O_A and unit vectors $\mathbf{x}_A, \mathbf{y}_A, \mathbf{z}_A$. As you can see on the figure 2b, the kinematic and dynamic equations of the AUV can be expressed with respect to the universal reference frame $\{U\}$ and also to the body frame $\{B\}$. The point Q depicted on the figure 2b corresponds to the center of mass of the AUV and we suppose that



(a) INFANTE AUV



(b) AUV's model

Figure 2

it coincides with the point O_B, the origin of the body frame $\{B\}$. Moreover, the coordinates of the point Q with respect to the universal frame $\{U\}$ are $\mathbf{q} = (x, y, 0)^T$ (just remember that the $\mathbf{z}$-component of the coordinates is always zero). Furthermore, the yaw angle, which defines the rotation matrix from the frame $\{B\}$ to the frame $\{U\}$, is denoted ψ_B. Then according to the paper [**4**], we denote $\mathbf{v}_t = (u, v, 0)^T$ as the velocity of the point Q in the frame $\{U\}$ and it expressed in the coordinates of the body frame $\{B\}$. The velocity u is called the surge velocity, while the velocity v is called the sway velocity. We also assume that the surge velocity u is never equal to zero. By exploiting the notation above, the following kinematic equations of the AUV can be deduced:

$$\begin{aligned} \dot{x} &= v_t \cos(\psi_W) \\ \dot{y} &= v_t \sin(\psi_W) \\ \dot{\psi}_B &= r + \dot{\beta} \end{aligned} \tag{1}$$

where $\beta = \arctan(\frac{v}{u})$ is called the side-slip angle, $r = \dot{\psi}_B$ is the angular speed of the robot, $v_t = \|\mathbf{v}_t\| = \sqrt{u^2 + v^2}$. On the other hand, the dynamic model for the AUV can be expressed in the following manner:

$$\begin{aligned} F &= m_u \dot{u} + d_u \\ 0 &= m_v \dot{v} + m_{ur} u r + d_v \\ \Gamma &= m_r \dot{r} + d_r \end{aligned} \tag{2}$$

The dynamic model above involves the mass and the moment of inertia of the AUV and also classic hydrodynamic derivatives. For the ease of readability those quantities will not be described in this section and you can consult the paper [**4**] of Lapierre et al. in order to get explicit physical and mathematical definition of those quantities.

Moreover, we must ask ourselves which quantities the path-following controller must take into account in order for the vehicle to converge to and follow the path. Firstly, the path-following controller should compute the distance between the center of mass of the vehicle Q and the closest point P on the path and secondly, the controller should also compute the angle between the velocity vector $\mathbf{v}_t$ and the tangent to the path at the point P. It is clear that the path-following controller should reduce both these quantities to zero in order to complete the path-following task.

Henceforth, that is where the notion of the *Serret-Frenet* frame $\{F\}$, which moves along the path (by the way, this frame is also displayed on the figure 2b) and will be used for the development of kinematic model of the AUV. The paper [**4**] states that the frame $\{F\}$ will act as a sort of *virtual target vehicle*, that will move along the path and will be followed by the *real* AUV. However, the origin O_F of $\{F\}$ will be defined in a way different from the one that was described in the article [**5**] of Micaelli and Samson and the article [**3**] of Encarnaçao et al. In fact, in the articles [**5**] and [**3**] O_F is the closest point on the path to the vehicle, while in the article [**4**] of Lapierre et al. the position of the $O_F = P$ will be defined by the convenient path-following controller. By doing so another parameter for the controller design is added and as a result, we have an extra degree of freedom. In fact, this technique allows to the authors Lapierre et al. to remove the strict initial condition constraint, which has been imposed in the papers [**5**] and

[**3**] in order for the path-following controller to be successful. To be more precise, this initial condition requires the *initial position error of the autonomous vehicle to be smaller than the smallest radius of curvature present in the path.* It is clear that this initial condition constraint is quite restrictive for the real-world environment and fortunately, the authors Lapierre et al. have succeeded in removing this stringent condition.

With the introduction of the new frame $\{F\}$, we thus introduce new notations. The variable s denotes the signed curvilinear abscissa of $P = O_F$ along the path. It is clear that the point Q (the center of mass of the vehicle) can either be expressed in the coordinates of the frame $\{U\}$, which are $q = (x, y, 0)^T$, or in the coordinates of $\{F\}$, which are $(s_1, y_1, 0)$. We denote ψ_F as the angle, which defines the rotation matrix from $\{U\}$ to $\{F\}$ and we impose $\omega_F = \dot{\psi}_F$. Henceforth, we have the following equations:

$$\begin{aligned} \omega_F &= \dot{\psi}_F = c_c(s)\dot{s} \\ \dot{c}_c(s) &= g_c(s)\dot{s} \end{aligned} \tag{3}$$

where $c_c(s)$ defines the path curvature and $g_c(s) = \frac{dc_c(s)}{ds}$ its derivative. By applying this new notation, the new kinematic model of the AUV can be written in the frame $\{F\}$:

$$\begin{aligned} \dot{s}_1 &= -\dot{s}\,(1 - c_c y_1) + v_t \cos\psi \\ \dot{y}_1 &= -c_c \dot{s} s_1 + v_t \sin\psi \\ \dot{\psi} &= \omega_W - c_c \dot{s} \end{aligned} \tag{4}$$

where the new variables $\psi = \psi_W - \psi_F$ and $\omega_W = \dot{\psi}_W = r + \dot{\beta}$ are introduced.

3.2.2 Nonlinear kinematic controller of the AUV

In this section and the sections above we will consider the kinematic and dynamic path-following controllers, which have been developed in the paper [**4**] of Lapierre et al. Before that we are going to introduce a new variable:

$$\delta(y_1) = -\psi_a \frac{e^{2k_\delta y_1} - 1}{e^{2k_\delta y_1} + 1}$$

which is a desired approach angle of the vehicle and accepts $\psi_a \in]0, \pi/2[$ and $k_\delta > 0$ as input parameters. This approach angle is quite important for the path-following control, because it will take part in defining the movements of the vehicle during the path-approach process. The first step will be to develop the kinematic controller for the AUV by assuming that the surge velocity u equals to the desired velocity $u_d > 0$. Then by taking into account the newly developed kinematic controller, the authors Lapierre et al. in the article [**4**] are going to exploit a backstepping technique in order to develop nonlinear dynamic controller for the input variables F (force exerted along the longitudinal axis of the AUV) and Γ (torque generated around the vertical axis of the AUV) as it has been done in the papers [**5**] of Micaelli and Samson and [**3**] of Encarnaçao et al.

Let's consider the first proposition of the article [**4**], which proposes the nonlinear kinematic path-following controller of the AUV:

Proposition 1 (Kinematic Controller)**.** *Consider the kinematic model of an AUV described in 1 and 4. Let the approach angle $\delta(y_1)$ be defined as above. Assume that the surge velocity of the vehicle is such that $u = u_d > 0$ (e.g. always constant). Suppose that the path to be followed is parameterized by its curvilinear abscissa s, and assume that for each s the variables ψ, s_1, y_1, c_c are well defined. Then the kinematic control law*

$$U_{\text{kin}} = \begin{cases} r = \dot{\delta} - \dot{\beta} - k_1(\psi - \delta) + c_c(s)\dot{s} \\ \dot{s} = \cos\psi v_t + k_2 s_1 \end{cases} \tag{5}$$

(where k_1, k_2 are arbitrary positive constants) drives y_1, s_1, ψ asymptotically to zero.

Moreover, it is safe to say that if the quantities y_1, s_1, ψ tend asymptotically to zero, then the autonomous vehicle will converge to and follow the path.

3.2.3 Nonlinear dynamic controller for the AUV

The kinematic controller defined in the previous section is only appropriate for the kinematic model of the AUV. So now it is important to take into account the vehicle dynamics in order to finish the path-following controller, developed in the paper [**4**] of Lapierre et al. In this section the backstepping technique is exploited in order to deduce the dynamic controller from the kinematic controller of the previous section. It is useful to note that in the previous kinematic controller the total velocity $v_t(t)$ depended on the desired surge velocity u_d for the speed $u(t)$ and we assumed that during the whole path-following phase we had $u = u_d$, which is not true in general. That is why the dynamic controller should guarantee that the quantity $u(t) - u_d$ tends asysmptotically to zero. The following proposition from the paper [**4**] defines the dynamic control law for the input variables F and Γ:

Proposition 2 (Dynamic Controller)**.** *Consider the kinematic and dynamic models of an AUV described in 1 and 2, respectively, and the corresponding path-following model in 4. Let the approach angle $\delta(y_1)$ be defined as in above and let a desired speed profile $u_d > u_{min} > 0$ for $u(t)$ be given. Suppose the path to be followed is parameterized by its curvilinear abscissa s, and assume that for each s the variables $\psi, s_1, y_1, c_c, dc_c/ds$ are well defined. Then the dynamic control law*

$$U_{dyn} = \begin{cases} \Gamma = m_r\alpha_r - d_r \\ F = m_u\left(\dot{u}_d - k_4\left(u - u_d\right)\right) - d_u \\ \dot{s} = \cos\psi v_t + k_2 s_1 \end{cases} \tag{6}$$

where $\alpha_r = \ddot{\delta} - \ddot{\beta} - (k_1 + k_3)(\dot{\psi} - \dot{\delta}) - (k_5 + k_1 k_3)(\psi - \delta) + c_c\ddot{s} + \frac{dc_c}{ds}\dot{s}$, the coefficients k_i are arbitrary positive gains, d_r and d_u are sums of hydrodynamic coefficients, drives $y_1, s_1, u - u_d, \psi$ asymptotically to zero.

In order to prove that the path-following controller actually works and the vehicle really converges to and follows the path, the authors Lapierre et al. of the paper [**4**] have performed a simulation with given initial conditions for the AUV's model. As you can see on the figure 3a, the simulation with the initial conditions shows us that the vehicle really converges to and follows the path. Is is useful to remark that as depicted on the figure 3a the initial position error of the vehicle does not need to be smaller than the smallest radius of curvature present in the path in order for the vehicle to complete the path-following task as the contrary has been required before in the papers [**5**] of Micaelli and Samson and [**3**] of Encarnaçao et al.

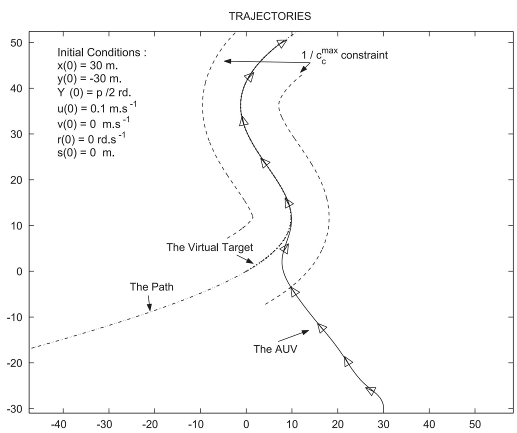
3.3 Redundant actuation system for an AUV

The following information about results and notations is based on the paper [**6**] of Lapierre et al.

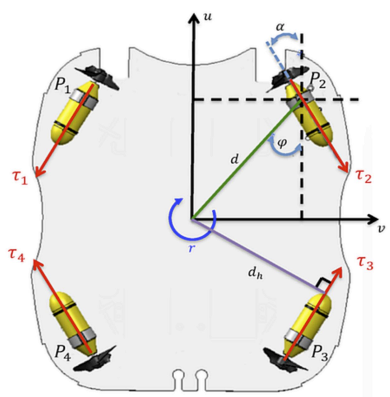
3.3.1 Context

The real-world environment in most cases imposes difficult conditions on autonomous vehicles. Moreover, the reactivity and actuators' saturation constraints and the minimization of the energy consumption are ones of the issues, which should be handled by the robotic system if it wants to not guarantee the successful completion of complex tasks, but also the completion of them in the best way possible, which is quite important given the fact that the robot's configuration should attract industrial companies in terms of their cost (construction, maintenance, etc.) and efficiency in order for the vehicle to be constructed in sufficient quantities and dedicated to the society's needs.

Furthermore, according to the article [**6**] the robotic system is said to be overactuated (in other words, redundant), if the number of actuators, that it possesses, is more than six *degrees of freedom* (DOFs) in case the $3D$ space is considered. As for the $2D$ space, the robotic system needs to have more actuators than three DOFs. The reason why the redundancy is very important for the autonomous systems is that it can be used for simultaneous control of



(a) Path-following simulation



(b) Jack system

Figure 3

several tasks. For example, it can be exploited for the path-following control and the obstacle avoidance for an AUV in an underwater environment. The pioneering research has been done by Hanafusa et al. in the paper [**11**], where the redundancy of the robotic system is exploited with the usage of Jacobian matrices and inverse kinematics and the order of priority is given to secondary tasks for robot manipulators.

3.3.2 Redundancy management

For the sake of simplicity let's place ourselves in the $2D$ space. The paper [**6**] uses the Jack robotic system, an AUV developed by the CISCREA company. As you can see on the figure 3b, the Jack system possesses four actuators, which is more than three degrees of freedom, and thus, the Jack system is overactuated in the $2D$ space. Now let's introduce the following notation of the paper [**6**]: $\mathbf{F}_B$ denotes the generated forces and torques of the vehicle with respect to the body frame $\{B\}$, e.g. $\mathbf{F}_B = [\mathbf{F}_u, \mathbf{F}_v, \Gamma_r]^T$ with the $\mathbf{F}_u$ being the surge force, $\mathbf{F}_v$ the sway force and Γ_r the torque generated around the vertical axis of the vehicle. Moreover, the four actuators of the Jack system exert four respective forces, which can be expressed in the following column vector: $\mathbf{F}_m = [\mathbf{F}_{m,1}, \mathbf{F}_{m,2}, \mathbf{F}_{m,3}, \mathbf{F}_{m,4}]^T$. Now we would like to build a relation between $\mathbf{F}_B$ and $\mathbf{F}_m$. By analyzing the positions $[d_{x,i}, d_{y,i}, d_{z,i}, \psi_{m,i}]^T$ of each motor depicted on the figure 3b, the article [**6**] of Lapierre et al. constructs the following algebraic relation between $\mathbf{F}_B$ and $\mathbf{F}_m$:

$$\mathbf{F}_B = A \cdot \mathbf{F}_m$$

where the actuation matrix A is a 3×4 constant matrix defined by the motor positions $[d_{x,i}, d_{y,i}, d_{z,i}, \psi_{m,i}]^T$. The redundancy management, which is based on this new relation, is discussed in the following sections and my solution, which generalizes the management of underactuated and overactuated system, is also presented in the sections below.

3.3.3 Saturation management

Even if dynamic path-following controller is well defined for an autonomous vehicle and the simulations show that this controller works very well in spite of initial conditions, the real world environment imposes harsh conditions in such way that the development of the path-following controller is not enough to guarantee the completion of the path-following task. In fact, for example, if the path-following controller requires one of the motors to exert the force $1000N$ and the maximum force that it can exert is in fact $200N$, then the path-following control will fail as the control inputs become saturated. That is why the saturation management is the crucial issue for the motion control of the robotic systems and it should be handled by any means necessary in order to guarantee the completion of the path-following task.

Actually the paper [**6**] of Lapierre et al. proposes a solution to resolve such issue. You can see on the figure 4 the relation between the motor characteristic $c_{m,i}$ and the exerted force $F_{m,i}$ for the Jack system. In fact, in reality the exerted force of the motor is commanded by its characteristic, e.g. in order for the motor to induce a force we should communicate to it a corresponding characteristic. As it can be seen on the figure 4, the *Dead zone* refers to certain characteristics, which does not induce any force for the corresponding motor. Henceforth, in order to minimize the energy consumption the *Dead zone* should be avoided for the robotic system and it is the part of the saturation problem. On the figure 4 you can see that the set $[c_m^{DZ-}, c_m^{DZ+}]$ corresponds to the characteristic inputs, which do not induce any force, and thus the corresponding quantities F_m^{DZ+} and F_m^{DZ-} are called the minimal attainable positive thrust and maximal negative thrust, respectively. Moreover, F_m^{max+} and F_m^{max-} denote the maximum positive thrust and minimum one, respectively.

In paper [**6**] of Lapierre et al. in order to resolve the issue of the *dead zone* two different solutions are proposed:

- Contraction of the *dead zone*: the characteristic inputs corresponding to the *dead zone* are made inaccessible and in such, the dead zone is truly avoided. However, such behaviour provokes oscillatory trend in characteristic inputs, inducing useless motor fatigue and the performance of the response depends on the motor reactivity.
- Compensation of the *dead zone* is achieved using a common motor regime applied to all horizontal actuators. On the contrary to the previous solution the oscillatory behaviour is removed

To get acquainted more in detail with two solutions above, I invite you to consult the paper [**6**] of Lapierre et al.

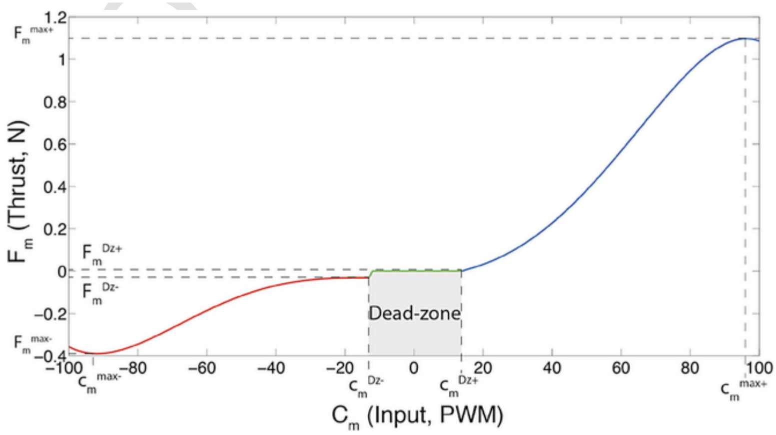


Figure 4: Relation between the motor characteristic and the exerted force

4 Obtained results

4.1 Generalization of the actuation management

Let's consider the overactuated system with the Jack system of the previous section, which has 4 motors. The paper [**6**] of Lapierre et al. on redundant actuation systems studies the problem of redundancy management with an actuation matrix A, which has a full rank. Now we would like to analyze the problem, where the action matrix A does not have a full rank in case one of our motors does not function anymore or some motors point in the same direction. Formally the problem of control can be presented as

$$\mathbf{F}_B = \mathbf{A} \cdot \mathbf{F}_m$$

where $\mathbf{F}_B = [F_u, F_v, \Gamma_r]^T$ (the generated forces and torques of the vehicle with respect to the body frame $\{B\}$) and $\mathbf{F}_m = [F_{m,1}, F_{m,2}, F_{m,3}, F_{m,4}]^T$ (the exerted forces of the motors) and A is a 3×4 actuation matrix. The notation is the same as in the previous section.

In case the actuation matrix A does not have a maximal rank, then for an $\mathbf{F_B}$ there exists $\mathbf{F_m}$ such that $\mathbf{F}_B = \mathbf{A} \cdot \mathbf{F}_m$ if and only if $\mathbf{F_B} \in range(A)$, where $range(A)$ is an image of matrix A. Let's suppose in following sections that $rank(A) = 2$. Then by the *Rank-nullity theorem* we have that $dim(Ker(A)) = 2$. It is clear that not all values of $\mathbf{F_B}$ can be achieved in our case as the actuation matrix A is not of a maximal rank. Then the set of possible values of $\mathbf{F_B}$ can be achieved through the computation of the basis of the range of the matrix A. By the *Gram-Schmidt process* each finite basis in an euclidian space can be converted to the new orthonormal basis. Thus, let's suppose that we have the orthonormal basis $\mathbf{B} = (e_1, e_2)$ of the range of A. Then the set of all possible values of $\mathbf{F_B}$ is $range(A) = \{\lambda \cdot e_1 + \mu \cdot e_2 : (\lambda, \mu) \in \mathbf{R}^2\}$. By doing so we get the constraints on the values of $\mathbf{F}_B$. Therefore, the set of all values of the desired resulting actions $\mathbf{F}_B^d$ must be the vector subspace $range(A)$. Even if A is not a full-rank matrix, it always has a unique Moore-Penrose inverse A^+ (for more information you can refer to the following source[7]). After that according to the same source we can conclude that $range(A) = ker(I - A \cdot A^+) = \{y = (A \cdot A^+) \cdot y : y \in \mathbf{R}^3\}$, because $I - A \cdot A^+$ is the orthogonal projector onto the kernel of A^T and we know that $ker(A^T)^\perp = range(A)$

To prove such property, firstly, the Moore-Penrose inverse A^+ can be computed through the rank decomposition of the matrix A. As $rank(A) = 2$, then we can compute the full-rank matrices B and C of sizes 3×2 and 2×4 respectively such that $A = B \cdot C$ and in this case, the pseudo-inverses B^+ and C^+ can be computed through the well-determined formula, e.g. $B^+ = (B^T B)^{-1} B^T$ and $C^+ = C^T (CC^T)^{-1}$. Thus, we have that $A^+ = C^+ \cdot B^+$. On the other hand, $I - A \cdot A^+$ is an orthogonal projector on $ker(A^T)$ and thus, $ker(I - A \cdot A^+) = ker(A^T)^\perp = range(A)$. Then we consider only the desired resulting actions $\mathbf{F}_B^d$, which satisfy the equation $(A \cdot A^+) \cdot \mathbf{F}_B^d = \mathbf{F}_B^d$. Thus, we get the new formulation of constraints on the values of $\mathbf{F}_B^d$.

Now let's take the desired resulting action $\mathbf{F}_B^d$ satisfying the equation $(A \cdot A^+) \cdot \mathbf{F}_B^d = \mathbf{F}_B^d$. The

[7] `https://en.wikipedia.org/wiki/Moore-Penrose_inverse`

main objective here is to compute the motor action $\mathbf{F_m}$ satisfying the equation $\mathbf{F}_B^d = \mathbf{A} \cdot \mathbf{F}_m$. Then the solution[8] of minimum norm to this equation is $\mathbf{F_m} = A^+ \cdot \mathbf{F}_B^d$ and the set of all solutions to the main equation is $\{A^+ \cdot \mathbf{F}_B^d + [I - A^+ \cdot A]\omega : \omega \in \mathbf{R}^4\}$, where $I - A^+ \cdot A$ is an orthogonal projector on $ker(A)$. We can easily verify that for all $\omega \in \mathbf{R}^4$,

$$A \cdot (A^+ \cdot \mathbf{F}_B^d + [I - A^+ \cdot A]\omega) = (A \cdot A^+) \cdot \mathbf{F}_B^d + (A - A \cdot A^+ \cdot A)\omega = \mathbf{F}_B^d + (A - A)\omega = \mathbf{F}_B^d$$

On the other hand, as $dim(ker(A)) = 2$ we can write down the set of all solutions to the main equation in an equivalent form, which is

$$A^+ \cdot \mathbf{F}_B^d + \lambda \cdot \mathbf{r_1} + \mu \cdot \mathbf{r_2} = \left[\mathbf{A}^+, \mathbf{r}_1, \mathbf{r_2}\right] \cdot \begin{bmatrix} \mathbf{F}_B^d \\ \lambda \\ \mu \end{bmatrix}$$

where $(\mathbf{r_1}, \mathbf{r_2})$ is an orthonormal basis of $ker(A)$ and $(\lambda, \mu) \in \mathbf{R}^2$.

This section actually shows that the result for the redundancy management announced in the paper [**6**] of Lapierre et al. can be actually generalized to underactuated robotic systems by taking into account the additional constraint on the set of the possible values for the desired resulting force vector $\mathbf{F}_B^d$

4.1.1 Example in 2D

Let's suppose that we have the *CUBE* robot in 2D, e.g. we have four actuators, and let's analyze the following configuration for the actuation matrix A:

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ -1 & 1 & -1 & 1 \end{bmatrix}$$

Then the set of the desired resulting actions $\mathbf{F}_B^d$ must satisfy the following constraint:

$$(I - A \cdot A^+) \cdot \mathbf{F}_B^d = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot \mathbf{F}_B^d = 0$$

and given the fact that $\mathbf{F}_B^d = [\mathbf{F}_u^d, \mathbf{F}_v^d, \mathbf{\Gamma}_r^d]^T$, then $\mathbf{F}_B^d$ must satisfy the constraint : $\mathbf{F}_v^d = 0$ corresponding to the null sway force.

In general, if the actuation matrix A of size $3 \times n$ does not have a maximal rank and is not null, e.g. $0 < rank(A) < 3$, then as $I - A \cdot A^+$ is the orthogonal projector onto $ker(A^T)$ and $dim(ker(A^T)) = dim(range(A)^\perp) = 3 - rank(A) \in \{1, 2\}$ by the $rank-nullity$ theorem, by the *spectral* theorem there exists an orthogonal matrix P of size 3 such that $I - A \cdot A^+ = P^T \cdot D \cdot P$,

[8] https://en.wikipedia.org/wiki/Moore-Penrose_inverse

where $D = diag(\underbrace{1, ..., 1}_{3-rank(A)}, 0, ..., 0)$. Henceforth, we get the equivalent formulation of the constraint: $P^T \cdot D \cdot P \cdot \mathbf{F}_B^d = 0$, which is equivalent to $D \cdot P \cdot \mathbf{F}_B^d = 0$ by multiplying P on the left by both sides. The new vector $P \cdot \mathbf{F}_B^d$ gives the change of coordinates for the forces in our system with each coordinate being the linear combination of the forces $\mathbf{F}_u, \mathbf{F}_v$ and $\mathbf{\Gamma}_r$. Moreover, with P being an orthogonal matrix the norm of the resulting forces is preserved after the change of coordinates, e.g. $||P \cdot \mathbf{F}_B^d|| = ||\mathbf{F}_B^d||$. Then the equation $D \cdot P \cdot \mathbf{F}_B^d = 0$ gives us $3 - rank(A)$ linearly independent combinations of the forces $\mathbf{F}_u, \mathbf{F}_v$ and $\mathbf{\Gamma}_r$, which are equal to zero. Thus, by doing so we get the equivalent set of constraints on the desired resulting actions for our system and moreover, we have explicitly deduced $3 - rank(A)$ directions in which the resulting force cannot be exerted. Finally, this knowledge of impossible dynamic directions will give us an important hint for the *path-following* control.

Now let's return to our example for the actuation matrix A. Actually the robotic system is said to be redundant if and only if the actuation matrix A have maximal rank. In our case, only the force in the v direction cannot be exerted, which means that our system is underactuated and we can still exert the forces $\mathbf{F}_u$ and $\mathbf{\Gamma}_r$. We can trace the parallel with the unicycle robot, discussed in the paper [**1**] of *Lapierre et al.* and which functioned in such way that the sway velocity v was always zero due to the friction constraint and thus, the sway force F_v was always zero, and deduce the dynamic and kinematic control for the $CUBE$ robot.

Moreover, if $rank(A) = 1$, then by performing the process described above we can prove that the robot is able to move only in one direction, which is quite restrictive for the *path-following* problem. Henceforth, we can confirm that the minimal rank of the actuation matrix A, for which we can guarantee the satisfaction of the *path-following* problem in the $2D$ space, is 2. It is useful to note that this case can be generalized in 2D space to the actuation matrix A of size $3 \times n$, where $n \geq 2$ is the number of actuators of our robot.

4.1.2 Case 3D

Let's refer again to the $CUBE$ robot in 3D space. As it has been shown in the previous section, by considering the actuation matrix of size $6 \times n$, where $n \geq 3$ is the number of actuators in our system, there exists an orthogonal matrix P of size 6 such that the set of the desired resulting actions $\mathbf{F}_B^d$ is represented by the linear system of equations: $D \cdot P \cdot \mathbf{F}_B^d = 0$, where $D = diag(\underbrace{1, ..., 1}_{6-rank(A)}, 0, ..., 0)$. As a result, we have $6 - rank(A)$ linearly independent combinations of the forces $\mathbf{F}_u, \mathbf{F}_v, \mathbf{F}_w, \mathbf{\Gamma}_r, \mathbf{\Gamma}_p, \mathbf{\Gamma}_q$, which must be equal to zero. Thus, we get in explicit manner $6 - rank(A)$ directions in which the resulting force cannot be exerted. It is safe to say that our system must have at least three degrees of freedom in order to guarantee the *path-following* constraint. Then we can suppose that $3 \leq rank(A) \leq 6$. If $rank(A) > 3$, then the *path-following* property can be satisfied and the results of the previous papers of Lapierre et al. can be exploited in this case. In case we have $rank(A) = 2$, then our system will be able to move only in a certain plan defined by two exerting forces, which are given by the constraints above. Moreover, if $rank(A) = 1$, then our robot will move only along a certain line in the space . As for the case when $rank(A) = 3$, we can deduce an example, where we can only exert torques $\mathbf{\Gamma}_r, \mathbf{\Gamma}_p, \mathbf{\Gamma}_q$ and thus, the robot becomes stationary. That is why the case when $rank(A) = 3$ should be considered in detail and we must analyze three generated possible directions, in which the vehicle can exert a resulting force, and conclude if the robotic system stays stationary or not according to these directions.

4.2 Selection criteria

Given the directions, in which our system can move, we now would like to determine if our system is well-posed, e.g. makes sense in the real life in terms of energy consumption, etc. For that let's consider a 2D case, where the actuation matrix A is

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 10^{-3} & 0 & 0 & 0 \\ -1 & 1 & -1 & 1 \end{bmatrix}$$

Then it is clear in this case that exerting a force in $\vec{v}$ direction costs too much in terms of energy and it would be disastrous to implement a *path-following* algorithm, which uses a non-zero force $\mathbf{F}_v$. That is why in this example the *path-following* algorithm must use only the forces $\mathbf{F}_u$ and $\mathbf{\Gamma}_r$, which is possible given to the recent papers.

4.2.1 First approach

In general, given the actuation matrix A, we have $rank(A)$ dynamic directions, in which our system can move. Moreover, these dynamic forces are linearly independent linear combinations of the initial force vectors. In order to estimate the cost of each direction in terms of energy consumption, the first approach to do so would be to refer ourselves to the fact that there exists an orthogonal matrix P of size 3 or 6 such that $D \cdot P \cdot \mathbf{F}_B^d = 0$, where $D = diag(\underbrace{1, ..., 1}_{n-rank(A)}, 0, ..., 0)$, where n is 3 or 6 depending on the dimension of our system. It is useful to note that $\mathbf{F}_B^d = A \cdot \mathbf{F}_m^d$, then we have $D \cdot P \cdot A \cdot \mathbf{F}_m^d = 0$. After having computed $P \cdot A$, we extract $rank(A)$ last lines of such matrix and we compute an euclidian norm of each line which will refer to the "energy" cost for each direction and such costs will be used for the comparison of the directions in terms of energy consumption.

4.2.2 Second approach

Let's place ourselves in a 2D example, where an actuation matrix A is

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 10^{-4} & 0 & 0 & 0 \\ -1 & 1 & -1 & 1 \end{bmatrix}$$

As it can be seen, the actuation matrix A is a full-rank matrix. Then the optimal solution $\mathbf{F}_m^d$ in terms of energy consumption is $A^+ \cdot \mathbf{F}_B^d$. We then can compute the cost of energy consumption for each force $\mathbf{F}_u, \mathbf{F}_v, \mathbf{\Gamma}_r$ from the Moore-Penrose pseudo-inverse A^+. For example, the cost of the force $\mathbf{F}_u$ is $c(\mathbf{F}_u) = 0,6$, the cost of $\mathbf{F}_v$ is $c(\mathbf{F}_v) = 14000$ and the cost of $\mathbf{\Gamma}_r$ is $c(\mathbf{\Gamma}_r) = 0,6$. Henceforth, exert a force in the $\vec{v}$ is very costly and must always be avoided in a path-following algorithm, giving priority to the directions $\vec{u}$ and $\vec{r}$. To sum up, the problem is not well-posed and such configuration of actuators must be modified.

On the other hand, we also have in general $A^+ \cdot \mathbf{F}^d_B = A^+ \cdot A \cdot \mathbf{F}^d_m$. As the matrix $A^+ \cdot A$ is an orthogonal projector onto the range of A^T and $rank(A^T) = rank(A) = 3$, there exists an orthogonal matrix Q of size 4 such that $A^+ \cdot A = Q^T \cdot diag(0,1,1,1) \cdot Q$. In our example, we have that the matrix Q is

$$\begin{bmatrix} -9.8 \cdot 10^{-18} & 0.7071 & -1.7 \cdot 10^{-16} & -0.7071 \\ -0.5395 & 0.3566 & -0.6472 & 0.3566 \\ 0.8178 & 0.3566 & -0.2772 & 0.3566 \\ 0.2002 & -0.4956 & -0.6845 & 0.4956 \end{bmatrix}$$

Henceforth, we get that

$$A^+ \cdot \mathbf{F}^d_B = A^+ \cdot A \cdot \mathbf{F}^d_m = Q^T \cdot diag(0,1,1,1) \cdot Q \cdot \mathbf{F}^d_m = \begin{bmatrix} 0 & -0.5395 & 0.8178 & 0.2002 \\ 0 & 0.3566 & 0.3566 & -0.4956 \\ 0 & -0.6472 & -0.2772 & -0.6845 \\ 0 & 0.3566 & 0.3566 & 0.4956 \end{bmatrix} \cdot Q \cdot \mathbf{F}^d_m = V \cdot Q \cdot \mathbf{F}^d_m$$

Then the first element of $Q \cdot \mathbf{F}^d_m$ does not give any contribution to the problem as the first column of the matrix V is zero. The first element in $Q \cdot \mathbf{F}^d_m$ refers to the direction formed by linear combination of the actuator forces and this direction will not give any gain to our path-following problem and will be omitted in $A^+ \cdot \mathbf{F}^d_B$, which is a solution of a minimum norm for the path-following problem. Given the fact that $A^+ \cdot \mathbf{F}^d_B$ is a solution of minimum norm for the path-following problem, it will eradicate the linear combinations of the actuator forces, which are useless to our problem.

4.2.3 Final approach: Eigen values approach

Let's consider the symmetric matrix $A \times A^T$. By supposing that A is of maximal rank, e.g. its rows are linearly independent, then the matrix $A \times A^T$ is invertible. Moreover, by the *spectral theorem* there exists an orthogonal matrix V of size n, which is 3 or 6 depending on the dimension of our space, and a diagonal matrix $D = diag(\lambda_1, ..., \lambda_n)$ such that $A \times A^T = V \times D \times V^T$ with $\lambda_i \neq 0$ given that the matrix $A \times A^T$ is invertible. It is useful to note that the columns v_i of the matrix V refer to the unitary eigen vectors with eigen value λ_i, then let's denote $V = (v_1, ..., v_n)$. Furthermore, the solution to the path-following problem with minimum energy consumption is $\mathbf{F}^d_m = A^+ \times \mathbf{F}^d_B$ and $A^+ = A^T \times (A \times A^T)^{-1}$ given the fact that A has linearly independent rows. Henceforth, $\mathbf{F}^d_m = A^T \times (A \times A^T)^{-1} \times \mathbf{F}^d_B = A^T \times V \times D^{-1} \times V^T \times \mathbf{F}^d_B = A^T \times V \times diag(\lambda_1^{-1}, ..., \lambda_n^{-1}) \times (\mathbf{F}^d_B)_{Base(v_1,...,v_n)}$, where $(\mathbf{F}^d_B)_{Base(v_1,...,v_n)}$ is coordinates of $\mathbf{F}^d_B$ in the base $(v_1, ..., v_n)$. By denoting $A = (a_1, ..., a_m)$ with a_i being $n \times 1$ vector column and referring to the force $\mathbf{F}^i_m$, we have the following result: $A^T \times V = (< a_i, v_j >)_{i,j}$, where each line refers to the coordinates of a_i in the base $(v_1, ..., v_n)$. Finally, we get that $\mathbf{F}_m^d = (< a_i, v_j > \times\lambda_j^{-1})_{i,j} \times (\mathbf{F}_B^d)_{Base(v_1,...,v_n)}$. In order to compute the cost of each direction given by $(\mathbf{F}_B^d)_{Base(v_1,...,v_n)}$, we just compute the euclidian norm of each column of the matrix $(< a_i, v_j > \times\lambda_j^{-1})_{i,j}$, which is proportional to the quantities λ_j^{-1} given that by the Cauchy-Schwarz inequality we have for all j, $\sum_{i=1}^{m} \frac{<a_i,v_j>^2}{\lambda_j^2} \leq \sum_{i=1}^{m} \frac{||a_i||^2||v_j||^2}{\lambda_j^2} = \frac{1}{\lambda_j^2}\sum_{i=1}^{m} ||a_i||^2$. Then we note that these maximum bounds can be compared using the eigen values λ_j, which will give an estimation of the cost of each direction of $(\mathbf{F}_B^d)_{Base(v_1,...,v_n)}$. Henceforth, we can conclude that infinitely small values of λ_j will engender considerable costs for certain directions. By doing so, we eliminate the directions, which have infinitely high costs. We must also note that the newly generated directions are already orthogonal, because the new basis is orthonormal, and thus we do not need to search for the directions orthogonal to the *expensive* directions.

To sum up, it is sufficient to compute the eigen values and the corresponding eigen vectors of the full-rank symmetric matrix $A \times A^T$. Then we eliminate the eigen vectors corresponding to infinitely small eigen values and the new path-following algorithm will be based on the eigen vectors, which were spared after this process.

4.3 Saturation of the actuators

As it has been stated before the issue of the actuators saturation must be resolved by any means necessary in order to guarantee the completion of the path-following task and minimize the energy consumption.

In this section we will suppose that the actuation matrix A is full-rank. It is important to note that in real life the motors placed on the robot possess a maximum capacity, which in turn will place a certain limitation on the path-following algorithm. Moreover, we must not forget the notion of the *Dead Zone*, which means that there is the minimum resulting force to be exerted in order to make robot move. Henceforth, the set of all possible values for $\mathbf{F}_m$ is of a form $I_1 \times ... \times I_m$ with $I_i = [F_{i,min}^-; F_{i,max}^-] \cup [F_{i,min}^+; F_{i,max}^+]$. Then we can conclude that the set of values of $\mathbf{F_m}$ is union of product of compact real intervals. As a result, the set of the possible values for the $\mathbf{F}_m$ is $\bigcup_{i=1}^{2^m} K_i$, where K_i is a product of compact intervals. Then the possible values for the $\mathbf{F}_B$ is $\bigcup_{i=1}^{2^m} A \cdot K_i$, where $A \cdot K_i = \{A \cdot x; x \in K_i\}$, which in turn is a compact and convex set as an image of the compact convex set K_i by linear application.

Let's imagine that the robot is following a path using the path-following algorithm. Then in the process we are given a certain desired resulting force $\mathbf{F}_B^d$ and thus, must generate a certain actuation force $\mathbf{F}_m^d$ by taking into account the limitations of the actuators, i.e. the generated $\mathbf{F}_m^d$ must belong to the set $I_1 \times ... \times I_m$. In order to simplify the computations, let's place ourselves in the $2D$ space, i.e. $n = 3$. Then as our system is redundant, the number of the actuators m is bigger than n. Moreover, as the matrix A is full-rank, then by the *rank-nullity theorem* we have that $dim(Ker(A)) = m - rank(A) = m - 3$. Then be noting that $I - A^+ \cdot A$ is the orthogonal projector onto $ker(A)$, given the desired resulting force $\mathbf{F}_B^d$ a feasible actuation force $\mathbf{F}_m^d$ will be of the form $A^+ \cdot \mathbf{F}_B^d + (I - A^+ \cdot A) \cdot x \in I_1 \times ... \times I_m$ with $x \in \mathbf{R}^m$. Then the main problem is to find a certain vector $x \in \mathbf{R}^m$ so that the desired actuation force $\mathbf{F}_m^d$ becomes *feasible*, i.e. belongs to the set $I_1 \times ... \times I_m$. If $A^+ \cdot \mathbf{F}_B^d$ already belongs to the set $I_1 \times ... \times I_m$ then our problem is resolved. In the other case, we must find a vector $x \in \mathbf{R}^m$, which satisfies the constraint: $A^+ \cdot \mathbf{F}_B^d + (I - A^+ \cdot A) \cdot x \in I_1 \times ... \times I_m$, which is equivalent to find a vector in the set $Ker(A) \cap (J_1 \times ... J_m)$, where $J_1 \times ... \times J_m = (I_1 \times ... \times I_m) - A^+ \cdot \mathbf{F}_B^d = (I_1 - \{A^+ \cdot \mathbf{F}_B\}_1) \times ... \times (I_m - \{A^+ \cdot \mathbf{F}_B\}_m)$, given the fact that the matrix $I - A^+ \cdot A$ is an orthogonal projector onto $Ker(A)$.

From now on we suppose that $\mathbf{F}_B^d$ is fixed and that $A^+ \cdot \mathbf{F}_B^d$ does not belong to the set $I_1 \times ... \times I_m$. Then our main objective is to find a vector $r \in \mathbf{R^m}$ such that $A \cdot r = 0$ and $r \in J_1 \times ... \times J_m$ and thus, our final solution will be $\mathbf{F}_m^d = A^+ \cdot \mathbf{F}_B^d + r$ with an energy norm $||\mathbf{F}_m^d||^2 = ||A^+ \cdot \mathbf{F}_B^d||^2 + ||r||^2$. Henceforth, in order to satisfy a minimum energy constraint we are interested in finding a feasible solution r, which has a minimum norm.

Moreover, in order to satisfy the minimize the energy consumption, we should solve the following *quadratic programming* (QP) problem:

$$\begin{aligned} \min_{r} \quad & \|A^+ \cdot \mathbf{F}_B^d + r\|^2 \\ \text{s.t.} \quad & r \in J_1 \times ... \times J_m \\ & A \cdot r = 0 \end{aligned} \tag{7}$$

What we have in the equation above is not formally the *quadratic programming* problem, because the set over which we want to minimize our target function is $Ker(A) \cap J_1 \times ... \times J_m$ and it is not convex. In other words, what we want to get in the end is the *convex optimization* problem, i.e. the minimization of a convex function over the convex set. On the other hand, the function which we want to minimize is $\|A^+ \cdot \mathbf{F}_B^d + r\|^2 = \langle A^+ \cdot \mathbf{F}_B^d + r, A^+ \cdot \mathbf{F}_B^d + r\rangle = \|A^+ \cdot \mathbf{F}_B^d\|^2 + \|r\|^2 + 2\langle A^+ \cdot \mathbf{F}_B^d, r\rangle = \|A^+ \cdot \mathbf{F}_B^d\|^2 + r^T \cdot r + (2A^+ \cdot \mathbf{F}_B^d)^T \cdot r = \|A^+ \cdot \mathbf{F}_B^d\|^2 + (1/2) r^T \cdot P \cdot r + q^T \cdot r$ with $q = 2A^+ \cdot \mathbf{F}_B^d$ and $P = 2 \cdot I_m$ with I_m being an identity matrix of size m. As the quantity $A^+ \cdot \mathbf{F}_B^d$ is fixed, then it is equivalent to minimize the function $r \mapsto (1/2) r^T \cdot P \cdot r + q^T \cdot r$. Furthermore, it is already useful to note that the function $r \mapsto \|A^+ \cdot \mathbf{F}_B^d + r\|^2$ is not only convex (which implies that any local minimum, if it exists, is a global minimum), but also strictly convex, i.e. it possesses at most one global minimum. Henceforth, the function $r \mapsto (1/2) r^T \cdot P \cdot r + q^T \cdot r = \|A^+ \cdot \mathbf{F}_B^d + r\|^2 - \|A^+ \cdot \mathbf{F}_B^d\|^2$ is also strictly convex given that the quantity $A^+ \cdot \mathbf{F}_B^d$ is fixed. On the contrary, as it has been mentioned above, the set $ker(A) \cap J_1 \times ... \times J_m$ over which we want to minimize is not convex. But we already know that $J_1 \times ... \times J_m = I_1 \times ... \times I_m - A^+ \cdot \mathbf{F}_B^d = \bigcup_{i=1}^{2^m} K_i - A^+ \cdot \mathbf{F}_B^d = \bigcup_{i=1}^{2^m} (K_i - A^+ \cdot \mathbf{F}_B^d)$, where the K_i is cartesian product of compact real intervals and the sets K_i are disjoint. By posing $K_i - A^+ \cdot \mathbf{F}_B^d = C_i$ for all i, where C_i is also a cartesian product of compact real intervals and the sets C_i are also disjoint.

Now let's fix C_i. As C_i is a cartesian product of compact intervals, then there exists a matrix G_i of size $2m \times m$ and a vector b_i of $\mathbf{R}^{2m}$ such that $r \in C_i$ is equivalent to $G_i \cdot r \leq b_i$. Then we can define formally the following *quadratic programming* subproblem (called $QP(i)$), which is:

$$\begin{aligned} \min_{r} \quad & (1/2)r^T \cdot P \cdot r + q^T \cdot r \\ \text{s.t.} \quad & G_i \cdot r \leq b_i \\ & A \cdot r = 0 \end{aligned} \tag{8}$$

Moreover, in the equation above the set over which we want to minimize our strictly convex target function is $Ker(A) \cap C_i$, which is convex compact. Then as the target function $r \mapsto (1/2)r^T \cdot P \cdot r + q^T \cdot r$ is continuous, then it reaches its bounds over the set $Ker(A) \cap C_i$. Henceforth, the subroblem *QP(i)* will have a solution (which will be unique if it exists as the target function is strictly convex) if and only if the set $Ker(A) \cap C_i$ is not empty.

Finally, we implement an iteration by solving the quadratic programming problem *QP(i)* for each $i \in \{1, ..., 2^m\}$ and over the course of this loop we update the solution r_{sol} if and only if the new minimum value of the target function becomes smaller. It is also possible that for all i, the set $Ker(A) \cap C_i$ is empty, which implies that our general problem does not possess the solution and the configuration of the actuators must be modified in order to satisfy the path-following objective or the value of the desired resulting force $\mathbf{F}_B^d$ must be changed.

As for the time complexity of the subproblem $QP(i)$ for each i, we can base our reasoning on the paper [**14**] of Vavasis. In this paper the author discusses about the time complexity of the quadratic programming algorithm. To be more precise, the author analyzes two cases: convex optimization and non-convex optimization. As for each i, the subproblem $QP(i)$ is a *convex optimization QP* problem, then we can restrict ourselves only to the first case, discussed in the paper of Vavarsis. According to the paper [**14**], the best-known time complexity for the convex *Quadratic Programming* problem is based on the work provided by the paper [**13**] of Renegar, which proposes a solution to improve the time complexity of the following *Linear Programming* problem:

$$\begin{aligned} \max_{x} \quad & c \cdot x \\ \text{s.t.} \quad & Bx \geq k \end{aligned} \tag{9}$$

where B is the matrix of size $n \times m$ and $k \in R^n$, $x, c \in R^m$. So for our case as the dimension of the forces' space n is either 3 or 6, the best-known time complexity for the convex *Quadratic Programming* subproblem $QP(i)$, which is based on the work provided by the paper [**13**] written by Renegar, can be solved in $O(m^{1/2}L)$ iterations, where L is the number of digits required to write (P, q, G_i, b_i, A), with each iteration requiring $O(m^3)$ arithmetic operations. Henceforth, the total time complexity of the subproblem $QP(i)$ is $O(m^{1/2}Lm^3) = O(m^{3.5}L)$. Moreover, in our case L is proportional to m^2, so the total time complexity for the subproblem $QP(i)$ is $O(m^{5.5})$. Finally, as we perform 2^m iterations in order to solve each subproblem $QP(i)$ for $i \in \{1, ..., 2^m\}$, the total time complexity of our algorithm is $O(2^m m^{5.5})$.

Moreover, we should remember that we are considering at the moment the *CUBE* system upon which these solutions will apply and we know that that the number of actuators in this system is $m = 8$ and even if we suppose that other group of extra actuators will be added to the

$CUBE$ system, the total number of actuators m will never be significant given the fact that in the contrary case the path-following controller will require much more computations and the construction and the maintenance of the system will be very costly. That is why even if the theoretical complexity of the process above is exponential, it will not be costly when put in practice.

In order to go further in the acceleration of the process above, it is useful to note that these iterations, which exploit the QP solver, are independent as the convex sets C_i are all disjoint, and thus, these iterations can be distributed between the multiple parallel processes available on the machine. In other words, in order to accelerate the execution of our algorithm we can use *Parallel programming* and all iterations of the loop will be executed independently on the available processes.

4.3.1 Reactivity constraint

Apart from the saturation constraints on the actuators, we would like to satisfy the reactivity constraints. In other words, what we want is to make the actuation forces produce the maximum yield factor. It means that if we take a look at the figure 4, then the points of the reactivity refer to the points, where the derivative of the graph becomes maximum in absolute value. In our case, there will be two points of the reactivity on the left and the right of the vertical $y - axis$ and we will denote them $\mathbf{F}^{\pm}_{react}$. Henceforth, we can refer ourselves to the previous section in order to satisfy the reactivity constraints. As we want the actuation forces, which are exerted by the motors, to produce the maximum yield factor and as the saturation constraints persist and they also must be satisfied, then we can formulate our problem in the same way as it has already been done in the equation (1):

$$\begin{aligned} \min_{r} \quad & \|A^{+} \cdot \mathbf{F}^{d}_{B} + r - \mathbf{F}^{\pm}_{react}\|^2 \\ \text{s.t.} \quad & r \in J_1 \times ... \times J_m \\ & A \cdot r = 0 \end{aligned} \tag{10}$$

The equation 10 will be resolved separately for $\mathbf{F}^{+}_{react}$ and $\mathbf{F}^{-}_{react}$ for a fixed $\mathbf{F}^{d}_{B}$ by *Quadratic Programming* and only the solution (if a solution exists), which minimizes the objective function the most, will be retained. The objective function in the equation above stays strictly convex as in the previous section.

Let's consider the equation 10 for $\mathbf{F}^{+}_{react}$ and the corresponding problem will be called (10+). It is useful to note that by applying the same reasoning as in the previous section, the solution to the equation (4+) exists if and only if the set $Ker(A) \cap (J_1 \times \ldots \times J_m)$, over which the objective function is minimized, is not empty. Then we can conclude that the solution to the equation (10+) exists if and only if the saturation constraints can be satisfied, i.e. the equation 8 has a solution. The same reasoning can be applied to the equation (10−). Henceforth, instead of trying to satisfy only the saturation constraints we can try to resolve the equation 10 in order to satisfy the saturation and reactivity constraints at the same time. This choice really depends on the user, who can decide by keeping in mind his needs if the reactivity constraint is really important for the *path-following problem* and if it is the case, then the user will try to solve directly the equation 10. Moreover, it can been seen easily that the resolution by *Quadratic Programming* of the equation 10 has the same time and space complexity as the equation 8.

We can easily imagine that in reality that the *path-following problem* has constraints other than reactivity and saturation, that we will need to satisfy. By taking into account the saturation of the actuators, which will always persist whatever the problem, and by denoting $\mathbf{F}^{opt}$ the force criteria of the considered constraint, which should be as near as possible to the actuation forces, then we can formulate the problem, which we want to solve, in the following way:

$$\begin{aligned} \min_{r} \quad & \|A^{+} \cdot \mathbf{F}_{B}^{d} + r - \mathbf{F}^{opt}\|^{2} \\ \text{s.t.} \quad & r \in J_1 \times ... \times J_m \\ & A \cdot r = 0 \end{aligned} \tag{11}$$

We can always apply the *Quadratic Programming* in order to solve the equation above and proceed to the same reasoning explained in the previous sections.

5 Conclusion

To sum up, we have analyzed the *path-following* motion control by considering both kinematic and dynamic control laws. More and more researchers concentrate their efforts on this type of motion control, because they believe that it will bring more efficient solutions and is simpler to be exploited than the other two techniques: *point stabilization* and *trajectory tracking.* Moreover, the redundant autonomous systems have been proved be useful for the completion of several tasks at the same time. The notion of task priority, explained in the paper [**11**] of Nakamura et al., can be exploited in order to profit from the redundancy of the system and complete secondary tasks.

Moreover, the issue of the actuators' saturation of an autonomous vehicle should not be ignored given the fact that the saturation can cause the path-following controller to fail as the input variables will not be able to satisfy the quantitative demand at some point of time. The solutions have been brought to the generalization of actuation management in the robotic systems in $2D$ and $3D$ space, to the selection criteria of directions to which the path-following controller should give priority in the first place and finally, to the satisfaction of the saturation and reactivity constraints (and other constraints if the user deems it necessary to add them for analysis).

In extend the areas of my research, I think it would be appropriate to dedicate more time to the *task priority* notion, explained in the paper [**11**] of Nakamura et al., in order to develop kinematic and dynamic models, which will guarantee the completion of the path-following task, that will be the primary task, and the completion of other secondary tasks (for example, keep the orientation of the surge force in direction to the fixed point in space). I would also like to see the application of my solutions in the construction of the $CUBE$ system, which is said to be finished later this year and to be tested near the source of Lez also this year. I hope that my solutions will make a good fit in the project *Lez 2020* and will be put in practice by my supervisor Lionel Lapierre.

References

[1] Lapierre L., Soetanto D., Pascoal A., *Non-singular path following control of a unicycle in the presence of parametric modeling uncertainties*, WILEY International Journal of Robust and Nonlinear Control, 16 :10, 485-503, 10.1002/rnc.1075, 2006

[2] Samson, C., Ait-Abderrahim, K. (1991). *Mobile robot control. Part 1: Feedback control of a non-holonomic mobile robots (Technical Report No. 1281).* Sophia-Antipolis, France: INRIA

[3] P. Encarnação, A. Pascoal, M. Arcak, *Path Following for Autonomous Marine Craft*, IFAC Proceedings Volumes, Volume 33, Issue 21, 2000

[4] L. Lapierre, D. Soetanto, *Nonlinear path-following control of an AUV*, Ocean Engineering, Volume 34, Issues 11–12, 2007, Pages 1734-1744, ISSN 0029-8018

[5] A. Micaelli, C. Samson, *Trajectory tracking for unicycle-type and two-steering-wheels mobile robots.* [Research Report] RR-2097, INRIA. 1993. ffinria-00074575f

[6] B. Ropars, L. Lapierre, A. Lasbouygues, D. Andreu, R. Zapata, *Redundant actuation system of an underwater vehicle*, Ocean Engineering, Volume 151, 2018, Pages 276-289, ISSN 0029-8018

[7] L. Lapierre, R. Zapata, P. Lepinay, B. Ropars, *Karst exploration: Unconstrained attitude dynamic control for an AUV*, Ocean Engineering, Volume 219, 2021, 108321, ISSN 0029-8018

[8] Lapierre, Lionel Indiveri, Giovanni. (2007). *Non-Singular Path-Following, Control of Wheeled Robots with velocity actuator saturations.* IFAC Proceedings Volumes. 6.10.3182/20070903-3-FR-2921.00006.

[9] Lapierre, Lionel. (2009). *Robust diving control of an AUV.* Ocean Engineering. 36.92-104.10.1016/j.oceaneng.2008.10.006.

[10] Thapa, Gyan. (2018). *On the Moore-Penrose generalized inverse matrix.* 95.

[11] Nakamura Y, Hanafusa H, Yoshikawa T. *Task-Priority Based Redundancy Control of Robot Manipulators.* The International Journal of Robotics Research. 1987;6(2):3-15. doi:10.1177/027836498700600201

[12] L. Lapierre, R. Zapata. *A guaranteed obstacle avoidance guidance system: The safe maneuvering zone.* Autonomous Robots, Springer Verlag, 2012, pp.177-187. ff10.1007/s10514-011-9269-5ff. ffhal-00733826f

[13] Renegar, J. *A polynomial-time algorithm, based on Newton's method, for linear programming.* Mathematical Programming 40, 59–93 (1988). https://doi.org/10.1007/BF01580724

[14] Vavasis, S.A. (2001). *Complexity Theory: Quadratic Programming.* In: Floudas, C.A., Pardalos, P.M. (eds) Encyclopedia of Optimization. Springer, Boston, MA. https://doi.org/10.1007/0-306-48332-$7_6$5