

# POLITECNICO DI TORINO

Master's Degree in Mechatronic Engineering



Master's Degree Thesis

## Modelling and Control of a Remotely Operated Vehicle (ROV)

Supervisors

Prof.ssa GIULIANA MATTIAZZO

Ing. FABIO CARAPELLESE

Candidate

SIMONE SCILLETTA

Luglio 2024



# Summary

The aim of this work is to analyze the crucial aspect of ROV dynamics and to develop and compare performance of different kind of control systems able to counteract hydrostatic forces, external disturbances and unbalance in order to let the underwater drone with stable attitude at a specific depth. We will describe the various mathematical models used for simulate the dynamic behavior of the ROV and for control purposes. Subsequently, different control techniques will be analyzed, and their performances will be compared. At the end of the work we will choose the control system to apply based on a trade off between performance, robustness, online computational complexity and design effort.

# Acknowledgements



# Table of Contents

<b>List of Tables</b>	VII
<b>List of Figures</b>	VIII
<b>Acronyms</b>	XI
<b>1 Introduction</b>	1
1.1 State of art . . . . .	1
1.2 Thesis motivation . . . . .	3
<b>2 ROV Modelling</b>	5
2.1 Simulation model: Fossen model . . . . .	5
2.1.1 Notation . . . . .	6
2.1.2 Rigid Body's Kinematics . . . . .	7
2.1.3 Rigid Body's Dynamics . . . . .	10
2.1.4 Hydrostatics . . . . .	16
2.1.5 Hydrodynamics . . . . .	18
2.2 System parameters . . . . .	21
2.2.1 Thruster Dynamics, TCM and Control Allocation . . . . .	23
2.2.2 Hydrodynamic parameters . . . . .	26
2.3 Control-oriented model . . . . .	29
2.3.1 Heave model . . . . .	30
2.3.2 Pitch model . . . . .	31
2.3.3 Roll model . . . . .	32
<b>3 Control System Design</b>	34
3.1 Control design via pole placement . . . . .	36
3.1.1 Anti wind-up for saturation handling . . . . .	40
3.1.2 Heave control via pole placement . . . . .	41
3.1.3 Pitch control via pole placement . . . . .	42
3.1.4 Roll control via pole placement . . . . .	44

3.2	Control design through LQR . . . . .	46
3.3	MPC design . . . . .	50
<b>4</b>	<b>Control Performance Comparison</b>	<b>57</b>
4.1	Performance comparison on simulator . . . . .	57
4.2	Critical analisys . . . . .	62
	<b>Bibliography</b>	<b>66</b>

# List of Tables

2.1	The notation of SNAME (1950) for marine vessels . . . . .	6
2.2	EVA's parameters . . . . .	22
2.3	Moment arms of 8 thrusters of EVA . . . . .	24



# List of Figures

1.1	Different ROVs . . . . .	2
1.2	ROVs with various tools . . . . .	2
1.3	PoliTOcean Student Team . . . . .	4
1.4	EVA . . . . .	4
2.1	Reference frames. Source: Handbook of Marine Craft Hydrodynamics and Motion Control [4] . . . . .	8
2.2	Body-fixed frame of the ROV . . . . .	9
2.3	Kinematics of a particle of the rigid body . . . . .	11
2.4	Gravitational and buoyancy forces acting on the center of gravity (CG) and center of buoyancy (CB) of a submarine. Source: Handbook of Marine Craft Hydrodynamics and Motion Control [4] .	16
2.5	Rigid-body and fluid kinetic energy. Source: Handbook of Marine Craft Hydrodynamics and Motion Control [4] . . . . .	19
2.6	EVA schematic of thrust forces w.r.t. CG . . . . .	23
2.7	T200 BlueRobotics Thrust (kgf) with respect to the ESC PWM Input Value ( $\mu s$ ) . . . . .	26
2.8	Simplified CAD of EVA . . . . .	27
2.9	Mesh of the ROV, side section view . . . . .	28
2.10	Mesh of the ROV, top section view . . . . .	28
2.11	Bode Diagram of Heave model . . . . .	30
2.12	Bode Diagram of Pitch model . . . . .	31
2.13	Bode Diagram of Roll model . . . . .	32
3.1	1 DOF architecture . . . . .	34
3.2	Control problem identification . . . . .	36
3.3	2 DOF architecture . . . . .	37
3.4	Feedback form of biproper controller . . . . .	40
3.5	Anti wind-up control loop . . . . .	41
3.6	Pole placement control for heave dynamic . . . . .	42
3.7	Pole placement regulation for pitch dynamic . . . . .	44

3.8	Pole placement tracking for pitch dynamic . . . . .	44
3.9	Pole placement control for roll dynamic . . . . .	45
3.10	Static state feedback control architecture . . . . .	47
3.11	LQR with set point tracking structure . . . . .	47
3.12	LQR for heave dynamic . . . . .	48
3.13	LQR regulation for pitch dynamic . . . . .	49
3.14	LQR tracking for pitch dynamic . . . . .	49
3.15	LQR for roll dynamic . . . . .	50
3.16	MPC for heave dynamic . . . . .	55
3.17	MPC regulation for pitch dynamic . . . . .	55
3.18	MPC tracking for pitch dynamic . . . . .	56
3.19	MPC for roll dynamic . . . . .	56
4.1	Heave tracking performance comparison . . . . .	58
4.2	Pitch tracking performance comparison . . . . .	59
4.3	Heave disturbance rejection performance comparison . . . . .	60
4.4	Pitch disturbance rejection performance comparison . . . . .	60
4.5	Roll disturbance rejection performance comparison . . . . .	61
4.6	Critical comparison of controllers . . . . .	62



# Acronyms

**AUV**

Autonomous Underwater Vehicles

**BF**

Body Frame

**CFD**

Computational Fluid Dynamic

**CNC**

Computerized Numerical Control

**CG**

Center of Gravity

**DCM**

Direct Cosine Matrix

**ECEF**

Earth-centered Earth-fixed

**HDPE**

High Density PolyEthylene

**IF**

Inertial Frame

**IMU**

Inertial Measurement Unit

**LQR**

Linear Quadratic Regulator

**MATE**

Marine Advanced Technology Education

**MPC**

Model Predictive Control

**NED**

North-East-Down

**PCB**

Printed Circuit Board

**QP**

Quadratic Programming

**ROV**

Remotely Operated Vehicles

**SNAME**

Society of Naval Architects and Marine Engineers

**TCM**

Thrust Configuration Matrix

# Chapter 1

## Introduction

The underwater world has always posed a challenge for humans due to its inhospitable and inaccessible nature. However, advancements in technology have opened up new frontiers, allowing humans to venture into the depths of the ocean without physically entering the water. One remarkable innovation in this field is the Remotely Operated Vehicle (ROV), a robotic system specifically designed for underwater operations. This thesis will address the topic of mathematical modeling and control of a ROV. After a brief introduction on the state of art and the motivations of the thesis, two mathematical modeling techniques will be discussed: one for simulation purposes, and another aimed for designing control laws. Various control techniques will be developed and their performances will be compared.

### 1.1 State of art

ROVs are unmanned, remote-controlled submersible robots that enable humans to access and interact with the underwater world from the safety of the surface. These highly sophisticated machines consist of various components, including a control system, propulsion system, manipulator arms, cameras, and sensors. The control system allows operators to maneuver the ROV in three dimensions, providing real-time visuals and data feedback [1].

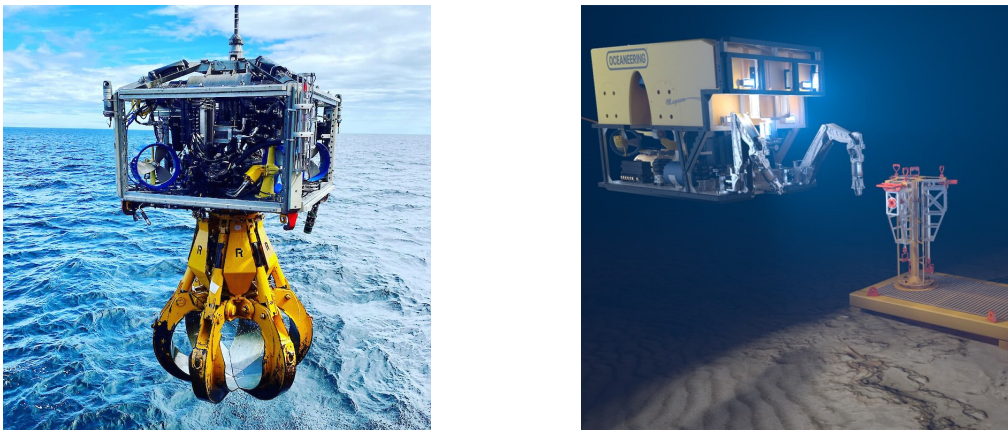
ROVs find applications in a wide range of industries and fields. In the offshore oil and gas industry, they play a crucial role in underwater inspection, maintenance, and repair of subsea infrastructure. ROVs are also extensively used in underwater exploration and research, enabling scientists to study marine life, geological formations, and ecosystems that were previously inaccessible. Moreover, they are employed in marine archaeology to explore and document submerged archaeological sites.

Environmental monitoring and conservation efforts greatly benefit from the use of ROVs, allowing scientists to assess the health of marine ecosystems, study endangered species, and identify potential pollution sources. In search and rescue operations, ROVs aid in locating and retrieving objects or individuals in challenging underwater environments. Additionally, these versatile robots assist in the maintenance and monitoring of subsea installations, such as underwater cables, pipelines, and offshore wind farms. ROVs are also frequently deployed by military organizations to enhance their capabilities in surveillance and search and rescue operations, all while maintaining cost efficiency [2].



**Figure 1.1:** Different ROVs

The development of ROVs can be traced back to the mid-20th century when the first prototypes emerged. Early versions were primitive and limited in capabilities. Over time, advancements in materials, control systems, and imaging technology have propelled the evolution of ROVs. Significant milestones include the introduction of tethered ROVs, the development of work-class ROVs capable of heavy-duty tasks, and the integration of advanced sensors and manipulator arms [3].



**Figure 1.2:** ROVs with various tools

In conclusion, ROVs have revolutionized the way humans explore and interact with the underwater realm. All the studies present in the literature on the control

of an ROV are based on mathematical models by Fossen [4] and those that do not is because they decide to use black box models based on System Identification, as seen in [5][6][7][8]. For control, the most commonly used technique is tuning a simple PID controller as seen in [9][10][7][8], before moving on to nonlinear controls such as sliding mode control as done in [11][12][10][13][14]. The problem with PID is that it may not be able to reject disturbances, while the issue with sliding mode control is that exact knowledge of the ROV's mathematical model is required, which is difficult due to the large number of parameters that need to be estimated. The objective of this thesis is therefore to implement various control techniques with specific requirements to overcome the issues outlined.

## 1.2 Thesis motivation

The aim of this thesis is the control of the ROV EVA, a prototype developed by the PoliTOcean team at the Polytechnic University of Turin. The PoliTOcean team has been involved in underwater robotics for several years and participates annually in the World Championship MATE (Marine Advanced Technology Education) ROV Competition in the USA. The team focuses on prototyping underwater vehicles such as ROVs and floats. The EVA prototype is equipped with 8 thrusters (T200 by BlueRobotics [15]), four for vertical movement and four for maneuvering. The overall power supply is 48V and is provided through the tether into the electronics box, where the Power Printed Circuit Board (PCB) distributes it across all the other boards. The ROV is equipped with various sensors, including an Inertial Measurement Unit (IMU) with an integrated Kalman Filter (WT61p by WitMotion [16]) that can directly provide roll and pitch values, in addition to linear accelerations and angular velocities, and a barometer (Bar02 by BlueRobotics [17]). These are the sensors that will detect measurements useful for control.

Furthermore, EVA has two camera boxes, one at the front and one at the bottom, and a two-degree-of-freedom robotic arm with which it can perform operations at depth. The frame is made of high-density polyethylene (HDPE), while the electronics box, made of an aluminum alloy, is enclosed by very thick plexiglass.

The entire system is designed to withstand depths close to 300 meters and is controlled by a joystick from the surface. All electronic components and PCBs are manufactured by the Electronic Division, the firmware is entirely written by the Firmware Division, and the mechanics are designed and manufactured by the Mechanical Division, also using 3D printing and CNC machining (Computerized Numerical Control). Additionally, the team includes the Hydrodynamic Division, which deals with Computational Fluid Dynamics (CFD) for calculating hydrodynamic parameters useful for the ROV's mathematical model, and the Control System Division, which handles the design of control systems.





**Figure 1.3:** PoliTOcean Student Team

Given that the ROV must be capable of performing manipulations at various depths, it is essential that it be easily maneuverable. The ROV is designed to be positive buoyant, and as it must manipulate various objects, it is continuously subjected to disturbances that alter its attitude. In addition, the system is not perfectly balanced and therefore requires attitude compensation. Therefore, to successfully carry out operations, the presence of a control system capable of maintaining the ROV at the desired depth and with a stable attitude is crucial. It must also be capable of counteracting disturbances. The purpose of this thesis is precisely the development of the control system that automatically commands the thrust and stabilizes EVA to the desired depth and attitude.



**Figure 1.4:** EVA

# Chapter 2

## ROV Modelling

In this chapter different mathematical models of the ROV are presented. The first model, discussed in section 2.1, is a high-fidelity model used to simulate its dynamics, while the other models, presented in section 2.3, are control-oriented models, representing simplified dynamics of the ROV. They are useful because they are linear models and can be expressed in state space and transfer function form, and therefore they are directly applicable for controller design.

### 2.1 Simulation model: Fossen model

In this section the high fidelity ROV mathematical model is derived. This model will be used for simulation purpose, since it describes exactly the dynamic behaviour of our ROV to be controlled. Modelling of rigid bodies moving in a fluid has been widely studied in literature. Please, refer to [4] and [18] for more details.

The study of dynamics can be divided into two parts: kinematics, which treats only geometrical aspects of motion, and dynamics, which is the analysis of the forces causing the motion.

The overall 6 DoF marine craft equation of motion can be written in a vectorial setting according to [4], hence the following relation holds:

$$\dot{\boldsymbol{\eta}} = \mathbf{J}_{\Theta}(\boldsymbol{\eta})\boldsymbol{\nu} \quad (2.1)$$

$$\mathbf{M}\dot{\boldsymbol{\nu}} + \mathbf{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \mathbf{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \mathbf{g}(\boldsymbol{\eta}) = \boldsymbol{\tau} \quad (2.2)$$

In equation (2.1)  $\mathbf{J}_{\Theta}$  is the Jacobian matrix. For what regards equation (2.2) we have that:  $\mathbf{M}$  is the sum of the rigid body mass matrix  $\mathbf{M}_{RB}^{CG}$  and the added mass matrix  $\mathbf{M}_A$ ,  $\mathbf{C}(\boldsymbol{\nu})$  is the sum of the rigid body Coriolis and centrifugal contribution  $\mathbf{C}_{RB}^{CG}$  and added mass Coriolis and centrifugal contribution  $\mathbf{C}_A(\boldsymbol{\nu})$ ,  $\mathbf{D}(\boldsymbol{\nu})$  includes all damping effects and  $\mathbf{g}(\boldsymbol{\eta})$  is the hydrostatic term.

DoF	Forces and moments	Linear and angular velocities	Positions and Euler angles
Surge	$X$	$u$	$x$
Sway	$Y$	$v$	$y$
Heave	$Z$	$w$	$z$
Roll, Heel	$K$	$p$	$\phi$
Pitch, Trim	$M$	$q$	$\theta$
Yaw	$N$	$r$	$\psi$

**Table 2.1:** The notation of SNAME (1950) for marine vessels

### 2.1.1 Notation

The motion of a ROV in 6 DoFs can be represented in a vectorial form using the SNAME notation (Society of Naval Architects and Marine Engineers - 1950) in table 2.1 where six individual coordinates are generalised to describe the position and orientation and their time derivatives describe the linear and angular velocities of the vehicle.

According to the SNAME notation (SNAME 1950), the generalised pose, velocity and forces and moments coordinates can be addressed by (2.3), (2.4) and (2.5) vectors, respectively.

$$\boldsymbol{\eta} = [x \ y \ z \ \phi \ \theta \ \psi]^T \quad (2.3)$$

$$\boldsymbol{\nu} = [u \ v \ w \ p \ q \ r]^T \quad (2.4)$$

$$\boldsymbol{\tau} = [X \ Y \ Z \ K \ M \ N]^T \quad (2.5)$$

Their sub-vectors are given by using the following vector notations:

- Position:

$$\boldsymbol{p} = [x \ y \ z]^T \in \mathbb{R}^3$$

- Euler angles:

$$\boldsymbol{\Theta} = [\phi \ \theta \ \psi]^T \in SO(3)$$

- Linear velocity:

$$\boldsymbol{v} = [u \ v \ w]^T \in \mathbb{R}^3$$

- Angular velocity:

$$\boldsymbol{\omega} = [p \ q \ r]^T \in \mathbb{R}^3$$

- Force on ROV:

$$\mathbf{f} = [X \ Y \ Z]^T \in \mathbb{R}^3$$

- Moment on ROV:

$$\mathbf{m} = [K \ M \ N]^T \in \mathbb{R}^3$$

where  $\mathbb{R}^3$  denotes the three dimensional Euclidean space and  $SO(3)$  indicates the three dimensional sphere in which three angles are defined on the interval of  $[-\pi, \pi]$  for  $\phi$  and  $\psi$ , and the interval of  $[-\pi/2, \pi/2]$  for  $\theta$ .

Therefore, the general motion of a ROV in 6 DoFs can be described by the following vectors:

$$\boldsymbol{\eta} = \begin{bmatrix} \mathbf{p} \\ \boldsymbol{\Theta} \end{bmatrix} \in \mathbb{R}^3 \times SO(3) \quad (2.6)$$

$$\boldsymbol{\nu} = \begin{bmatrix} \mathbf{v} \\ \boldsymbol{\omega} \end{bmatrix} \in \mathbb{R}^6 \quad (2.7)$$

$$\boldsymbol{\tau} = \begin{bmatrix} \mathbf{f} \\ \mathbf{m} \end{bmatrix} \in \mathbb{R}^6 \quad (2.8)$$

where  $\boldsymbol{\eta}$  is the position and orientation (pose) vector,  $\boldsymbol{\nu}$  is the linear and angular velocity vector and  $\boldsymbol{\tau}$  is the force and moment vector.

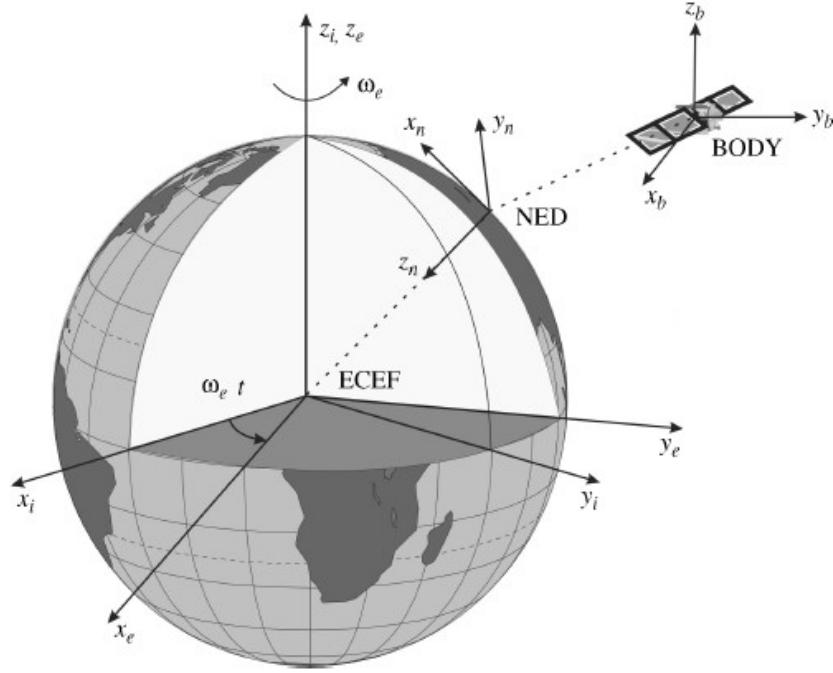
### 2.1.2 Rigid Body's Kinematics

When analyzing the motion of marine craft in 6 DOF, it is convenient to define two important reference frames (Figure 2.1):

**NED:** The North-East-Down (NED) coordinate system  $\{n\} = (x_n, y_n, z_n)$  with origin  $o_n$  is defined relative to the Earth's reference ellipsoid (World Geodetic System, 1984). It is defined as the tangent plane on the surface of the Earth moving with the craft. 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.

**BODY:** The Body-fixed reference frame  $\{b\} = (x_b, y_b, z_b)$  with origin  $o_b$  is a moving coordinate frame fixed to the craft. We describe position and orientation of ROV relative to the inertial reference frame ( $\{n\}$ ) while linear and angular velocities are expressed in the body-fixed frame. The body axes  $x_b$ ,  $y_b$  and  $z_b$  coincide with the principal axes of inertia, and they are defined as (see Figure 2.2):

- $x_b$ : longitudinal axis



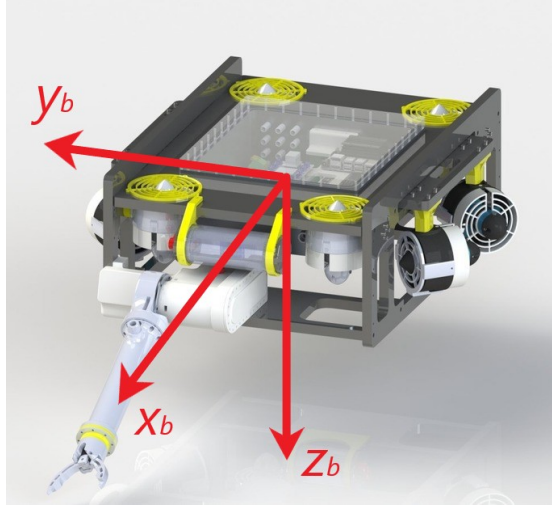
**Figure 2.1:** Reference frames. Source: Handbook of Marine Craft Hydrodynamics and Motion Control [4]

- $y_b$ : transversal axis
- $z_b$ : normal axis

It is possible to describe the orientation of the ROV with two different representation: Euler angles and quaternions. Depending on which of the two is used, we can derive the transformation between BODY and NED differently. In this work we will use Euler angles. The transformation of the ROV's kinematics variables from the Body-fixed frame  $\{b\}$  to the Inertial frame  $\{n\}$  can be expressed by a rotation matrix referred to as Direct Cosine Matrix (DCM). To derive the DCM it is possible to consider the total rotation in the 3-Dimensional space as a composition of three elementary rotations (Tait-Bryan 321 representation), each one of them described by one of the Euler angles. The final matrix representing the composition of the three rotations is:

$$\mathbf{R}_b^n(\Theta) = \mathbf{R}_z(\psi)\mathbf{R}_y(\theta)\mathbf{R}_x(\phi) \quad (2.9)$$

$$\mathbf{R}_b^n(\Theta) = \begin{bmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{bmatrix} \quad (2.10)$$



**Figure 2.2:** Body-fixed frame of the ROV

where we indicated the  $\sin(x)$  and  $\cos(x)$  functions as  $s_x$  and  $c_x$  respectively for abbreviation.

We can now define the equation that describe the transformation of the ROV's linear velocity from BODY to NED as:

$$\mathbf{v}^n = \mathbf{R}_b^n(\Theta)\mathbf{v}^b \quad (2.11)$$

Moreover, thanks to the fact that the DCM is an orthogonal matrix, we can get the inverse relation of (2.11) by simply remember that  $(\mathbf{R}_b^n(\Theta))^{-1} = (\mathbf{R}_b^n(\Theta))^T = \mathbf{R}_n^b(\Theta)$ .

As far as the transformation of the ROV's angular velocity is concerned, we need to highlight one important fact. The derivative of the Euler angles  $\dot{\Theta} = [\dot{\phi} \ \dot{\theta} \ \dot{\psi}]^T$  does not coincide with the actual angular velocity  $\boldsymbol{\omega}$  of the ROV. This is due to the fact that we use the Euler angles as representation of a generic rotation in the space. Several ways can be used to derive the Tait-Bryan 321 kinematic equation that represent the transformation of the ROV's angular velocity from the BODY frame to the NED.

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \frac{1}{c_\theta} \begin{bmatrix} c_\theta & s_\phi s_\theta & c_\phi s_\theta \\ 0 & c_\phi c_\theta & -s_\phi c_\theta \\ 0 & s_\phi & c_\phi \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} \iff \dot{\Theta} = \mathbf{T}_\Theta(\Theta)\boldsymbol{\omega}. \quad (2.12)$$

It is useful to collect the kinematic equations in 6-dimensional matrix forms. Recalling equations (2.6), (2.7), (2.11), (2.12), we can thus write [18] [4]:

$$\dot{\boldsymbol{\eta}} = \mathbf{J}_\Theta(\boldsymbol{\eta})\boldsymbol{\nu} \iff \begin{bmatrix} \dot{\mathbf{p}} \\ \dot{\Theta} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_b^n(\Theta) & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{T}_\Theta(\Theta) \end{bmatrix} \begin{bmatrix} \mathbf{v}^b \\ \boldsymbol{\omega}^b \end{bmatrix} \quad (2.13)$$

### 2.1.3 Rigid Body's Dynamics

Various methods can be explored in formulating the equations governing the motion of a rigid body. Below, we'll provide a concise overview of the Newton-Euler approach.

If  $\rho$  is the density of a particle of volume  $dV$  of a rigid body  $\mathcal{B}$ ,  $\rho dV$  is the total mass denoted by the position vector  $\mathbf{p}$  in an inertial frame  $O - xyz$ . If  $V_{\mathcal{B}}$  is the the total body volume we have:

$$m = \int_{V_{\mathcal{B}}} \rho dV \quad (2.14)$$

The center of mass of  $\mathcal{B}$  is defined as

$$\mathbf{p}_C = \frac{1}{m} \int_{V_{\mathcal{B}}} \mathbf{p} \rho dV \quad (2.15)$$

The linear momentum of the body  $\mathcal{B}$  is by definition the vector

$$\mathbf{l} = \int_{V_{\mathcal{B}}} \dot{\mathbf{p}} \rho dV = m \dot{\mathbf{p}}_C \quad (2.16)$$

Force is the derivative of linear momentum, so for a system with constant mass, the Newton's law of motion for the linear part is simply the Newton's equations of motion:

$$\mathbf{f} = \dot{\mathbf{l}} = m \frac{d}{dt} \dot{\mathbf{p}}_C \quad (2.17)$$

$$\mathbf{f} = m \ddot{\mathbf{p}}_C \quad (2.18)$$

where  $\mathbf{f}$  is the resultant of the external forces expressed in the Body-Frame  $\{b\}$  and  $\ddot{\mathbf{p}}_C$  is the acceleration of the center of gravity (CG).

For what regards attitude dynamics we have to consider a rigid body rotating with respect to an inertial reference frame with angular velocity  $\boldsymbol{\omega} = \omega_1 \mathbf{b}_1 + \omega_2 \mathbf{b}_2 + \omega_3 \mathbf{b}_3$ . With reference to the Figure 2.3, for a particle of mass  $m_i$  of the Body we can write:

$$\mathbf{p}_i = X \hat{\mathbf{i}}_1 + Y \hat{\mathbf{i}}_2 + Z \hat{\mathbf{i}}_3 \quad (2.19)$$

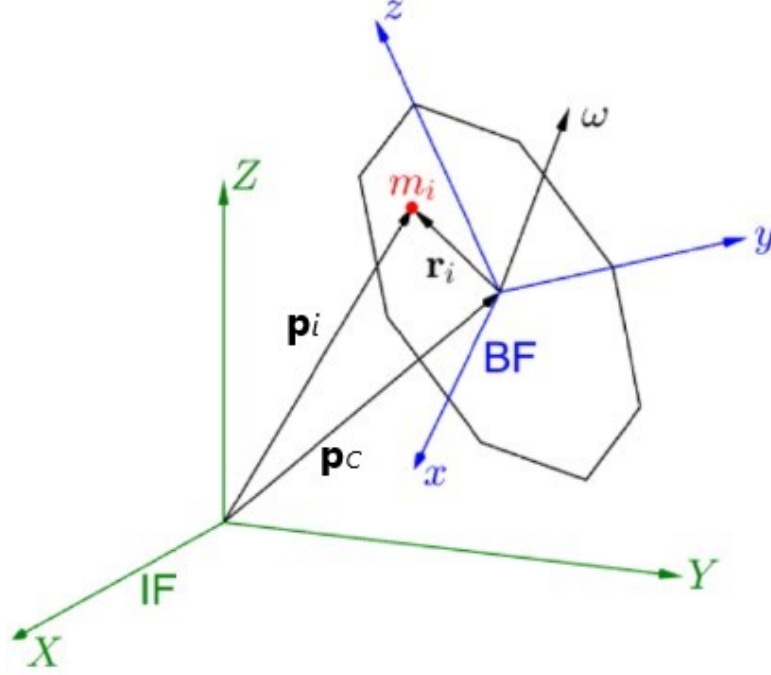
$$\mathbf{p}_C = X_o \hat{\mathbf{i}}_1 + Y_o \hat{\mathbf{i}}_2 + Z_o \hat{\mathbf{i}}_3 \quad (2.20)$$

$$\mathbf{r}_i = x \hat{\mathbf{b}}_1 + y \hat{\mathbf{b}}_2 + z \hat{\mathbf{b}}_3 \quad (2.21)$$

$$\dot{\mathbf{r}}_i^B = \dot{x} \hat{\mathbf{b}}_1 + \dot{y} \hat{\mathbf{b}}_2 + \dot{z} \hat{\mathbf{b}}_3 \quad (2.22)$$

$$\mathbf{p}_i = \mathbf{p}_C + \mathbf{r}_i \quad (2.23)$$

where  $\hat{\mathbf{i}}_j$ ,  $j = 1,2,3$  are the unit vectors of the Inertial Frame (IF) and  $\hat{\mathbf{b}}_j$ ,  $j = 1,2,3$  are the unit vectors of the Body Frame (BF).



**Figure 2.3:** Kinematics of a particle of the rigid body

We want now to derive equation (2.23) to get the velocity of the particle in the Inertial Frame. The derivative of the first term  $\mathbf{p}_C$  is just  $\dot{\mathbf{p}}_C$  as it is expressed in the IF, while the derivative of the second term  $\mathbf{r}_i$  is:

$$\frac{d\mathbf{r}_i}{dt} = \left( \frac{dx}{dt} \hat{\mathbf{b}}_1 + \frac{dy}{dt} \hat{\mathbf{b}}_2 + \frac{dz}{dt} \hat{\mathbf{b}}_3 \right) + \left( x \frac{d\hat{\mathbf{b}}_1}{dt} + y \frac{d\hat{\mathbf{b}}_2}{dt} + z \frac{d\hat{\mathbf{b}}_3}{dt} \right) \quad (2.24)$$

where the first bracket represents the variation of velocity in time and the second bracket is the variation of velocity due to the axes rotation, according to the Coriolis's Theorem. It is worth noting that, since we are referring to the Inertial Frame, the unit vectors  $\hat{\mathbf{b}}_j$ ,  $j = 1,2,3$  are considered non constants during the derivation as the Body Frame moves with respect to the Inertial Frame.



Recalling the Poisson's Theorem, we can state that there exists one and only one vector  $\boldsymbol{\omega}$  for which the following equations are valid:

$$\frac{d\hat{\mathbf{b}}_1}{dt} = \boldsymbol{\omega} \times \hat{\mathbf{b}}_1 \quad (2.25)$$

$$\frac{d\hat{\mathbf{b}}_2}{dt} = \boldsymbol{\omega} \times \hat{\mathbf{b}}_2 \quad (2.26)$$

$$\frac{d\hat{\mathbf{b}}_3}{dt} = \boldsymbol{\omega} \times \hat{\mathbf{b}}_3. \quad (2.27)$$

Plugging equations (2.25), (2.26) and (2.27) into equation (2.24) and arranging everything we can write the derivative of equation (2.23) as:

$$\dot{\mathbf{p}}_i = \dot{\mathbf{p}}_C + \dot{\mathbf{r}}_i^b + \boldsymbol{\omega} \times \mathbf{r}_i. \quad (2.28)$$

Moreover, it is worth keeping in mind the definition of Center of Mass (CG):

$$\mathbf{p}_C = \frac{1}{m} \sum_{i=1}^N m_i \mathbf{p}_i \quad (2.29)$$

where  $m = \sum_i m_i$  and  $N$  is the number of particles. We can rearrange (2.29) to get the following property:

$$\sum_{i=1}^N m_i (\mathbf{p}_i - \mathbf{p}_C) = \sum_{i=1}^N m_i \mathbf{r}_i = 0 \quad (2.30)$$

Now we can compute the Angular momentum (moment of momentum) of the particle as follows:

$$\mathbf{H}_i \doteq \mathbf{r}_i \times m_i \dot{\mathbf{p}}_i = \mathbf{r}_i \times m_i (\dot{\mathbf{p}}_C + \dot{\mathbf{r}}_i^b + \boldsymbol{\omega} \times \mathbf{r}_i). \quad (2.31)$$

Being  $\dot{\mathbf{r}}_i^B = 0$  (Rigid Body) and remembering this property of the cross product  $\mathbf{a} \times \mathbf{b} = -\mathbf{b} \times \mathbf{a}$  we can rewrite:

$$\mathbf{H}_i = \mathbf{r}_i \times m_i (\dot{\mathbf{p}}_C + \boldsymbol{\omega} \times \mathbf{r}_i) \quad (2.32)$$

$$= -\dot{\mathbf{p}}_C \times m_i \mathbf{r}_i + \mathbf{r}_i \times m_i (\boldsymbol{\omega} \times \mathbf{r}_i). \quad (2.33)$$

The angular momentum of the entire body is:

$$\mathbf{H} = -\sum_{i=1}^N \dot{\mathbf{p}}_C \times m_i \mathbf{r}_i + \sum_{i=1}^N \mathbf{r}_i \times m_i (\boldsymbol{\omega} \times \mathbf{r}_i) \quad (2.34)$$

$$= -\dot{\mathbf{p}}_C \times \sum_{i=1}^N m_i \mathbf{r}_i + \sum_{i=1}^N \mathbf{r}_i \times m_i (\boldsymbol{\omega} \times \mathbf{r}_i). \quad (2.35)$$

According to equation (2.30) it is possible to rewrite (2.35) as:

$$\mathbf{H} = \sum_{i=1}^N \mathbf{r}_i \times (\boldsymbol{\omega} \times \mathbf{r}_i) m_i \quad (2.36)$$

and for  $m_i \rightarrow dm$ , we can express everything in continuous form:

$$\mathbf{H} = \int_{V_B} \mathbf{r} \times (\boldsymbol{\omega} \times \mathbf{r}) dm. \quad (2.37)$$

By using the well known property of the cross product according to which

$$\mathbf{a} \times (\mathbf{b} \times \mathbf{c}) = (\mathbf{a} \cdot \mathbf{c}) \mathbf{b} - (\mathbf{a} \cdot \mathbf{b}) \mathbf{c} \quad (2.38)$$

we have that:

$$\mathbf{r} \times (\boldsymbol{\omega} \times \mathbf{r}) = (\mathbf{r} \cdot \mathbf{r}) \boldsymbol{\omega} - (\mathbf{r} \cdot \boldsymbol{\omega}) \mathbf{r} = (r^2 - \mathbf{r} \mathbf{r}^T) \boldsymbol{\omega}. \quad (2.39)$$

We can thus rewrite equation (2.37) as follows:

$$\mathbf{H} = \begin{bmatrix} I_x & I_{xy} & I_{xz} \\ I_{yx} & I_y & I_{yz} \\ I_{zx} & I_{zy} & I_z \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} = \mathbf{I} \boldsymbol{\omega} \quad (2.40)$$

where  $\mathbf{I}$  is the Inertia Matrix or Inertia Tensor,

$$I_x = \int_{V_B} (y^2 + z^2) dm \quad (2.41)$$

$$I_y = \int_{V_B} (x^2 + z^2) dm \quad (2.42)$$

$$I_z = \int_{V_B} (x^2 + y^2) dm \quad (2.43)$$

are referred to as moments of inertia and

$$I_{xy} = I_{yx} = - \int_{V_{\mathcal{B}}} xy \, dm \quad (2.44)$$

$$I_{xz} = I_{zx} = - \int_{V_{\mathcal{B}}} xz \, dm \quad (2.45)$$

$$I_{yz} = I_{zy} = - \int_{V_{\mathcal{B}}} yz \, dm \quad (2.46)$$

are referred to as products of inertia.

Suppose that a moment  $\mathbf{m}$  is acting on the body  $\mathcal{B}$ . The second law of dynamics for a rotating body is:

$$\dot{\mathbf{H}} = \mathbf{m}. \quad (2.47)$$

Hence we need to derive the equation (2.40). The inertia matrix is considered as a constant matrix while the derivative of  $\boldsymbol{\omega}$  in the Inertial Frame is:

$$\dot{\boldsymbol{\omega}} = \dot{\boldsymbol{\omega}}^b + \boldsymbol{\omega} \times \boldsymbol{\omega}^b \quad (2.48)$$

as for equation (2.24). Thus we get:

$$\dot{\mathbf{H}} = \mathbf{I}\dot{\boldsymbol{\omega}}^b + \boldsymbol{\omega} \times \mathbf{I}\boldsymbol{\omega}^b. \quad (2.49)$$

Replacing (2.49) in (2.47) we finally get the Euler's moment equation:

$$\mathbf{I}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \mathbf{I}\boldsymbol{\omega} = \mathbf{m}. \quad (2.50)$$

The latter equation is expressed in the Body-Frame  $\{b\}$  and the inertia matrix is computed about CG.

We can now derive the equation of dynamics computed with respect to the CG. From equation (2.18) it follows that:

$$\mathbf{f}^b = \frac{d}{dt} (m\mathbf{v}_C^b) = m (\dot{\mathbf{v}}_C^b + \boldsymbol{\omega}^b \times \mathbf{v}_C^b) \quad (2.51)$$

By remembering that the cross product operator is defined by:

$$\boldsymbol{\lambda} \times \mathbf{a} \doteq \mathbf{S}(\boldsymbol{\lambda})\mathbf{a} \quad (2.52)$$

where  $\mathbf{S}$  is a skew-symmetric matrix ( $\mathbf{S} = -\mathbf{S}^T$ ) defined as:

$$\mathbf{S}(\boldsymbol{\lambda}) = -\mathbf{S}^T(\boldsymbol{\lambda}) = \begin{bmatrix} 0 & -\lambda_3 & \lambda_2 \\ \lambda_3 & 0 & -\lambda_1 \\ -\lambda_2 & \lambda_1 & 0 \end{bmatrix}, \quad \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix} \quad (2.53)$$

with the off-diagonal elements that satisfy  $s_{ij} = -s_{ji}$  for  $i \neq j$  while the diagonal elements are zero, we can thus rewrite equations (2.50) and (2.51) as:

$$m \left[ \dot{\mathbf{v}}_C^b + \mathbf{S}(\boldsymbol{\omega}^b) \mathbf{v}_C^b \right] = \mathbf{f}^b \quad (2.54)$$

$$\mathbf{I} \dot{\boldsymbol{\omega}}^b - \mathbf{S}(\mathbf{I} \boldsymbol{\omega}) \boldsymbol{\omega}^b = \mathbf{m}^b. \quad (2.55)$$

The Newton–Euler equations (2.54) and (2.55) can be represented in matrix form according to

$$\mathbf{M}_{RB}^{CG} \dot{\boldsymbol{\nu}} + \mathbf{C}_{RB}^{CG} \boldsymbol{\nu} = \boldsymbol{\tau} \quad (2.56)$$

$$\begin{bmatrix} m \mathbf{1}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{v}}_C^b \\ \dot{\boldsymbol{\omega}}^b \end{bmatrix} + \begin{bmatrix} m \mathbf{S}(\boldsymbol{\omega}^b) & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & -\mathbf{S}(\mathbf{I} \boldsymbol{\omega}) \end{bmatrix} \begin{bmatrix} \mathbf{v}_C^b \\ \boldsymbol{\omega}^b \end{bmatrix} = \begin{bmatrix} \mathbf{f}^b \\ \mathbf{m}^b \end{bmatrix}. \quad (2.57)$$

We may want to derive the equations of motion with respect to an arbitrary origin CO to take advantage of ROV geometric properties. To do this, we can simply do a coordinate transformation of the equations of motion about CG to CO.

$$\mathbf{v}_C^b = \mathbf{v}_O^b + \boldsymbol{\omega}^b \times \mathbf{p}_C^b \quad (2.58)$$

$$= \mathbf{v}_C^b - \mathbf{p}_C^b \times \boldsymbol{\omega}^b \quad (2.59)$$

$$= \mathbf{v}_C^b + \mathbf{S}^T(\mathbf{p}_C^b) \boldsymbol{\omega}^b \quad (2.60)$$

It follows that

$$\begin{bmatrix} \mathbf{v}_C^b \\ \boldsymbol{\omega}^b \end{bmatrix} = \mathbf{H}(\mathbf{p}_C^b) \begin{bmatrix} \mathbf{v}_O^b \\ \boldsymbol{\omega}^b \end{bmatrix} \quad (2.61)$$

where  $\mathbf{H}(\mathbf{p}_C^b) \in \mathbb{R}^{3 \times 3}$  is a transformation matrix:

$$\mathbf{H}(\mathbf{p}_C^b) \doteq \begin{bmatrix} \mathbf{1}_{3 \times 3} & \mathbf{S}^T(\mathbf{p}_C^b) \\ \mathbf{0}_{3 \times 3} & \mathbf{1}_{3 \times 3} \end{bmatrix}, \quad \mathbf{H}^T(\mathbf{p}_C^b) = \begin{bmatrix} \mathbf{1}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{S}(\mathbf{p}_C^b) & \mathbf{1}_{3 \times 3} \end{bmatrix}. \quad (2.62)$$

Notice that angular velocity doesn't change. We now have to transform (2.57) from CG to CO using (2.61). This gives

$$\mathbf{H}^T(\mathbf{p}_C^b) \mathbf{M}_{RB}^{CG} \mathbf{H}(\mathbf{p}_C^b) \begin{bmatrix} \dot{\mathbf{v}}_O^b \\ \dot{\boldsymbol{\omega}}^b \end{bmatrix} + \mathbf{H}^T(\mathbf{p}_C^b) \mathbf{C}_{RB}^{CG} \mathbf{H}(\mathbf{p}_C^b) \begin{bmatrix} \mathbf{v}_O^b \\ \boldsymbol{\omega}^b \end{bmatrix} = \mathbf{H}^T(\mathbf{p}_C^b) \begin{bmatrix} \mathbf{f}^b \\ \mathbf{m}^b \end{bmatrix} \quad (2.63)$$

We can now define two new matrices in CO according to

$$\mathbf{M}_{RB}^{CO} \doteq \mathbf{H}^T(\mathbf{p}_C^b) \mathbf{M}_{RB}^{CG} \mathbf{H}(\mathbf{p}_C^b) \quad (2.64)$$

$$\mathbf{C}_{RB}^{CO} \doteq \mathbf{H}^T(\mathbf{p}_C^b) \mathbf{C}_{RB}^{CG} \mathbf{H}(\mathbf{p}_C^b) \quad (2.65)$$

These expressions can be expanded as [18] [4]:

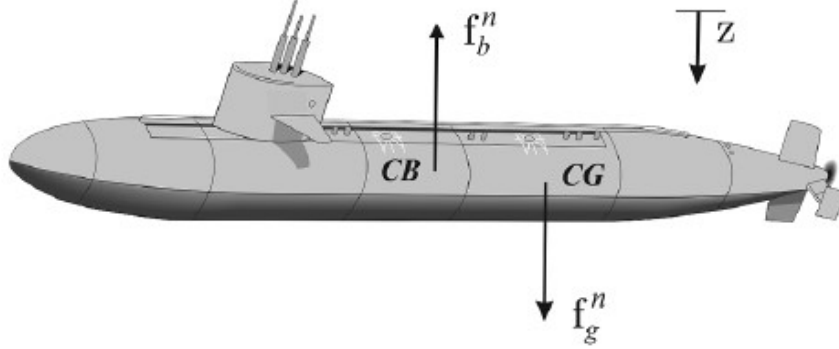
$$\mathbf{M}_{RB}^{CO} = \begin{bmatrix} m\mathbf{1}_{3 \times 3} & -m\mathbf{S}(\mathbf{p}_C^b) \\ m\mathbf{S}(\mathbf{p}_C^b) & \mathbf{I}_o \end{bmatrix} \quad (2.66)$$

$$= \begin{bmatrix} m & 0 & 0 & 0 & mz_C & -my_C \\ 0 & m & 0 & -mz_C & 0 & mx_C \\ 0 & 0 & m & my_C & -mx_C & 0 \\ 0 & -mz_C & my_C & I_x & -I_{xy} & -I_{xz} \\ mz_C & 0 & -mx_C & -I_{yx} & I_y & -I_{yz} \\ -my_C & mx_C & 0 & -I_{zx} & -I_{zy} & I_z \end{bmatrix} \quad (2.67)$$

$$\mathbf{C}_{RB}^{CO} = \begin{bmatrix} m\mathbf{S}(\boldsymbol{\omega}^b) & -m\mathbf{S}(\boldsymbol{\omega}^b) \mathbf{S}(\mathbf{p}_C^b) \\ m\mathbf{S}(\mathbf{p}_C^b) \mathbf{S}(\boldsymbol{\omega}^b) & -\mathbf{S}(\mathbf{I}_o \boldsymbol{\omega}^b) \end{bmatrix}. \quad (2.68)$$

#### 2.1.4 Hydrostatics

A rigid body submerged in a fluid is under the effect of gravitational force and buoyancy. Buoyancy is not function of the relative movement between body and fluid so it is an hydrostatic effect.



**Figure 2.4:** Gravitational and buoyancy forces acting on the center of gravity (CG) and center of buoyancy (CB) of a submarine. Source: Handbook of Marine Craft Hydrodynamics and Motion Control [4]

Let us define as

$$\mathbf{g}^n = \begin{bmatrix} 0 \\ 0 \\ 9.81 \end{bmatrix} m/s^2 \quad (2.69)$$

the acceleration of gravity,  $\Delta$  the volume of the body and  $m$  its mass. We can write the weight of the body as

$$\mathbf{W} = m\|\mathbf{g}^n\| \quad (2.70)$$

and its buoyancy as

$$\mathbf{B} = \rho\Delta\|\mathbf{g}^n\| \quad (2.71)$$

where  $\rho$  is the fluid density. Gravity acts in the center of mass  $\mathbf{p}_C^b = [x_C \ y_C \ z_C]^T$  while buoyancy acts in the center of buoyancy  $\mathbf{p}_B^b = [x_B \ y_B \ z_B]^T$  and them are represented in body-fixed frame by:

$$\mathbf{f}_G^b(\mathbf{R}_n^b) = \mathbf{R}_n^b \begin{bmatrix} 0 \\ 0 \\ W \end{bmatrix}, \quad (2.72)$$

$$\mathbf{f}_B^b(\mathbf{R}_n^b) = -\mathbf{R}_n^b \begin{bmatrix} 0 \\ 0 \\ B \end{bmatrix}. \quad (2.73)$$

We can now write a  $(6 \times 1)$  vector of force/moment in body-fixed frame:

$$\mathbf{g}_{RB}(\mathbf{R}_n^b) = - \begin{bmatrix} \mathbf{f}_G^b(\mathbf{R}_n^b) + \mathbf{f}_B^b(\mathbf{R}_n^b) \\ \mathbf{p}_C^b \times \mathbf{f}_G^b(\mathbf{R}_n^b) + \mathbf{p}_B^b \times \mathbf{f}_B^b(\mathbf{R}_n^b) \end{bmatrix}. \quad (2.74)$$

Expression (2.74) in terms of Euler angles is:

$$\mathbf{g}(\boldsymbol{\eta}_2) = \begin{bmatrix} (W - B) s_\theta \\ -(W - B) c_\theta s_\phi \\ -(W - B) c_\theta c_\phi \\ -(y_C W - y_B B) c_\theta c_\phi + (z_C W - z_B B) c_\theta s_\phi \\ (z_C W - z_B B) s_\theta + (x_C W - x_B B) c_\theta c_\phi \\ -(x_C W - x_B B) c_\theta s_\phi - (y_C W - y_B B) s_\theta \end{bmatrix}. \quad (2.75)$$

Usually a ROV is designed with  $B > W$  (positive buoyancy) such that the vehicle is able to surface automatically in case of power failure or other emergencies. Nevertheless, if the vehicle design is such that  $B \gg W$ , then we will need too much control energy to submerge the ROV. So a good design is a trade-off between control energy and positive buoyancy [18] [4].

### 2.1.5 Hydrodynamics

In this subsection, we will discuss the main hydrodynamic effects involved in the dynamics of a ROV. In particular, we will talk about the two most relevant effects: added mass and damping.

"The theory of fluidodynamics is rather complex and it is difficult to develop a reliable model for most of the hydrodynamic effects. A rigorous analysis for incompressible fluids would need to resort to the Navier-Stokes equations (distributed fluid-flow)."[18].

When a rigid body moves through a fluid, it experiences an additional inertia caused by the fluid surrounding it, which is accelerated by the body's movement. This phenomenon, known as added mass, must be taken into account because it becomes significant in underwater applications where the density of water (approximately  $\rho \simeq 1000\text{kg/m}^3$ ) is comparable to that of the ROV.

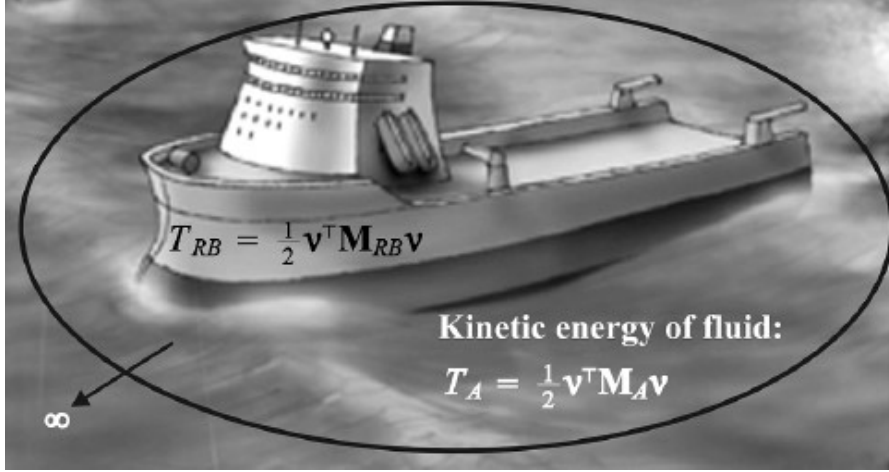
As the body moves through the fluid, it accelerates the fluid around it, requiring a force to achieve this acceleration. The fluid, in turn, exerts a reaction force equal in magnitude but opposite in direction. This reaction force constitutes the added mass contribution. Since any motion of the underwater craft induces movement in the otherwise stationary fluid, the fluid must displace and then re-fill the space behind the craft as it passes through. Consequently, the fluid gains kinetic energy that it wouldn't possess otherwise. (see Figure 2.5). The fluid kinetic energy  $T_A$  is written as:

$$T_A = \frac{1}{2} \boldsymbol{\nu}^T \mathbf{M}_A \boldsymbol{\nu}, \quad \dot{\mathbf{M}}_A = 0 \quad (2.76)$$

where  $\mathbf{M}_A = \mathbf{M}_A^T \geq 0$  is the  $6 \times 6$  *system inertia matrix* of added mass terms:

$$\mathbf{M}_A = - \begin{bmatrix} X_{\dot{u}} & X_{\dot{v}} & X_{\dot{w}} & X_{\dot{p}} & X_{\dot{q}} & X_{\dot{r}} \\ Y_{\dot{u}} & Y_{\dot{v}} & Y_{\dot{w}} & Y_{\dot{p}} & Y_{\dot{q}} & Y_{\dot{r}} \\ Z_{\dot{u}} & Z_{\dot{v}} & Z_{\dot{w}} & Z_{\dot{p}} & Z_{\dot{q}} & Z_{\dot{r}} \\ K_{\dot{u}} & K_{\dot{v}} & K_{\dot{w}} & K_{\dot{p}} & K_{\dot{q}} & K_{\dot{r}} \\ M_{\dot{u}} & M_{\dot{v}} & M_{\dot{w}} & M_{\dot{p}} & M_{\dot{q}} & M_{\dot{r}} \\ N_{\dot{u}} & N_{\dot{v}} & N_{\dot{w}} & N_{\dot{p}} & N_{\dot{q}} & N_{\dot{r}} \end{bmatrix} \quad (2.77)$$

The SNAME notation is used in this expression. Of course added mass introduce an added Coriolis and centripetal contribution. It can be proved that this effect can be parameterized as:



**Figure 2.5:** Rigid-body and fluid kinetic energy. Source: Handbook of Marine Craft Hydrodynamics and Motion Control [4]

$$\mathbf{C}_A(\boldsymbol{\nu}) = \begin{bmatrix} 0 & 0 & 0 & 0 & -a_3 & a_2 \\ 0 & 0 & 0 & a_3 & 0 & -a_1 \\ 0 & 0 & 0 & -a_2 & a_1 & 0 \\ 0 & -a_3 & a_2 & 0 & -b_3 & b_2 \\ a_3 & 0 & -a_1 & b_3 & 0 & -b_1 \\ -a_2 & a_1 & 0 & -b_2 & b_1 & 0 \end{bmatrix} \quad (2.78)$$

where

$$\begin{aligned} a_1 &= X_{\dot{u}}u + X_{\dot{v}}v + X_{\dot{w}}w + X_{\dot{p}}p + X_{\dot{q}}q + X_{\dot{r}}r \\ a_2 &= Y_{\dot{u}}u + Y_{\dot{v}}v + Y_{\dot{w}}w + Y_{\dot{p}}p + Y_{\dot{q}}q + Y_{\dot{r}}r \\ a_3 &= Z_{\dot{u}}u + Z_{\dot{v}}v + Z_{\dot{w}}w + Z_{\dot{p}}p + Z_{\dot{q}}q + Z_{\dot{r}}r \\ b_1 &= K_{\dot{u}}u + K_{\dot{v}}v + K_{\dot{w}}w + K_{\dot{p}}p + K_{\dot{q}}q + K_{\dot{r}}r \\ b_2 &= M_{\dot{u}}u + M_{\dot{v}}v + M_{\dot{w}}w + M_{\dot{p}}p + M_{\dot{q}}q + M_{\dot{r}}r \\ b_3 &= N_{\dot{u}}u + N_{\dot{v}}v + N_{\dot{w}}w + N_{\dot{p}}p + N_{\dot{q}}q + N_{\dot{r}}r \end{aligned} \quad (2.79)$$

Typically, when an underwater vehicle operates in 6 DOF at high speeds, its motion exhibits significant non-linearity and coupling. Nonetheless, in numerous applications involving AUV and ROV, movement is restricted to low speeds. Moreover, if the vehicle possesses three planes of symmetry, it implies that the influence stemming from the off-diagonal elements within the  $\mathbf{M}_A$  matrix can be disregarded.



So, the following expressions are obtained:

$$\mathbf{M}_A = \mathbf{M}_A^T = -diag \{X_{\dot{u}}, Y_{\dot{v}}, Z_{\dot{w}}, K_{\dot{p}}, M_{\dot{q}}, N_{\dot{r}}\}, \quad (2.80)$$

$$\mathbf{C}_A(\boldsymbol{\nu}) = \mathbf{C}_A^T(\boldsymbol{\nu}) = \begin{bmatrix} 0 & 0 & 0 & 0 & -Z_{\dot{w}}w & Y_{\dot{v}}v \\ 0 & 0 & 0 & Z_{\dot{w}}w & 0 & -X_{\dot{u}}u \\ 0 & 0 & 0 & -Y_{\dot{v}}v & X_{\dot{u}}u & 0 \\ 0 & -Z_{\dot{w}}w & Y_{\dot{v}}v & 0 & -N_{\dot{r}}r & M_{\dot{q}}q \\ Z_{\dot{w}}w & 0 & -X_{\dot{u}}u & N_{\dot{r}}r & 0 & -K_{\dot{p}}p \\ -Y_{\dot{v}}v & X_{\dot{u}}u & 0 & -M_{\dot{q}}q & K_{\dot{p}}p & 0 \end{bmatrix}. \quad (2.81)$$

For what regard damping effects, they are related to the presence of fluid viscosity that leads to the generation of dissipative drag and lift forces acting on the body. A typical simplification involves accounting for linear and quadratic damping terms and organizing them into a matrix,  $\mathbf{D}_{RB}$ , where  $\mathbf{D}_{RB}(\boldsymbol{\nu}) > \mathbf{0}$ ,  $\forall \boldsymbol{\nu} \in \mathbb{R}^6$ . These matrix coefficients are assumed to be constant.

$$\mathbf{D}_{RB}(\boldsymbol{\nu}) = -diag \{X_u, Y_v, Z_w, K_p, M_q, N_r\} + \quad (2.82)$$

$$- diag \{X_{u|u}|u|, Y_{v|v}|v|, Z_{w|w}|w|, K_{p|p}|p|, M_{q|q}|q|, N_{r|r}|r|\} \quad (2.83)$$

If we adopt a diagonal structure for the damping matrix, we are essentially disregarding the coupling dissipative terms. Viscous effects can be viewed as the combination of two forces: drag and lift forces. Drag forces are aligned with the relative velocity between the body and the fluid, whereas lift forces are perpendicular to it. It is assumed that both drag and lift forces act upon the body's center of mass. When considering a sphere in fluid motion, the drag force can be represented as:

$$F_{drag} = \frac{1}{2}\rho U^2 S C_d(R_n), \quad (2.84)$$

where  $\rho$  is the fluid density,  $U$  is the velocity of the sphere,  $S$  is the frontal area of the sphere,  $C_d$  is the drag coefficients and  $R_n$  is the Reynolds number. The lift forces act perpendicular to the direction of flow and can be represented as:

$$F_{lift} = \frac{1}{2}\rho U^2 S C_l(R_n, \alpha), \quad (2.85)$$

where  $C_l$  is the lift coefficient and  $\alpha$  is the angle of attack [4][18].

## 2.2 System parameters

For simulation purposes the high fidelity model discussed in section 2.1, was implemented in MATLAB Simulink to test the dynamic behaviour of EVA and different control systems performances. To do this we needed the numerical values for all the system parameters discussed before.

For what regard mechanical parameters we used different ways. For parameters like mass and Volume (useful for Bouyancy) we directly measured them experimentally, while for the Inertia Tensor and for the center of mass we referred to the complete CAD (Computer Aided design) of EVA realized from the mechanical division of PoliTOcean team.

The CAD is created in SolidWorks, and faithfully reproduces each component of the ROV, including electronics, and contains the properties of each of the materials comprising EVA We extracted the necessary information from the ‘SolidWorks mass property’ section of the software. Also the measurements necessary for computing the Thrust Configuration Matrix (TCM) used for control allocation were extracted through the CAD.

Hydrodynamic parameters such as added mass, damping and center of bouyancy were estimated from the hydrodynamic division of PoliTOcean team.

In the following subsections, the procedure for calculating the TCM, which is useful for control allocation, will be presented, along with the steps performed by the hydrodynamic division for calculating the hydrodynamic parameters.

All the parameters with their related values and measurement units are listed in table 2.2.

EVA's PARAMETERS		
$m$	mass	16.6 kg
$W$	weight	162.846 N
$B$	buoyancy	168.928 N
$I_x$	moment of inertia	0.40423 kg m <sup>2</sup>
$I_y$	moment of inertia	0.66054 kg m <sup>2</sup>
$I_z$	moment of inertia	0.92002 kg m <sup>2</sup>
$I_{xy}$	products of inertia	0.04893 kg m <sup>2</sup>
$I_{xz}$	products of inertia	0.06762 kg m <sup>2</sup>
$I_{yz}$	products of inertia	0.01505 kg m <sup>2</sup>
$ X_{\dot{u}} $	added mass	1.3169 kg
$ Y_{\dot{v}} $	added mass	2.5699 kg
$ Z_{\dot{w}} $	added mass	14.4260 kg
$ K_{\dot{p}} $	added mass	0.2089 kg m <sup>2</sup> rad <sup>-1</sup>
$ M_{\dot{q}} $	added mass	0.3637 kg m <sup>2</sup> rad <sup>-1</sup>
$ N_{\dot{r}} $	added mass	0.1482 kg m <sup>2</sup> rad <sup>-1</sup>
$ X_u $	linear damping	9.716 N s m <sup>-1</sup>
$ Y_v $	linear damping	20.53 N s m <sup>-1</sup>
$ Z_w $	linear damping	68.855 N s m <sup>-1</sup>
$ K_p $	linear damping	0.07 N s rad <sup>-1</sup>
$ M_q $	linear damping	0.07 N s rad <sup>-1</sup>
$ N_r $	linear damping	0.07 N s rad <sup>-1</sup>
$ X_{u u} $	quadratic damping	18.18 N s <sup>2</sup> m <sup>-2</sup>
$ Y_{v v} $	quadratic damping	21.66 N s <sup>2</sup> m <sup>-2</sup>
$ Z_{w w} $	quadratic damping	36.99 N s <sup>2</sup> m <sup>-2</sup>
$ K_{p p} $	quadratic damping	1.55 N s <sup>2</sup> rad <sup>-2</sup>
$ M_{q q} $	quadratic damping	1.55 N s <sup>2</sup> rad <sup>-2</sup>
$ N_{r r} $	quadratic damping	1.55 N s <sup>2</sup> rad <sup>-2</sup>
$\mathbf{p}_B^b$	coordinates of the CoB	[0; 0; -0.03] m
$\mathbf{p}_C^b$	coordinates of the CoM	[0; 0; 0] m

Table 2.2: EVA's parameters

### 2.2.1 Thruster Dynamics, TCM and Control Allocation

Since EVA has 8 thrusters, the thruster forces can be represented using a vector

$$\mathbf{F} = [F_1 \ F_2 \ F_3 \ F_4 \ F_5 \ F_6 \ F_7 \ F_8]^T, \quad (2.86)$$

while the control inputs can be represented using a vector

$$\mathbf{u} = [u_1 \ u_2 \ u_3 \ u_4 \ u_5 \ u_6 \ u_7 \ u_8]^T. \quad (2.87)$$

Given the force vector  $\mathbf{f} = [F_x \ F_y \ F_z]^T$  and the moment arms  $\mathbf{r} = [l_x \ l_y \ l_z]^T$ , the forces and moments in 6 DoFs can be determined by [19]:

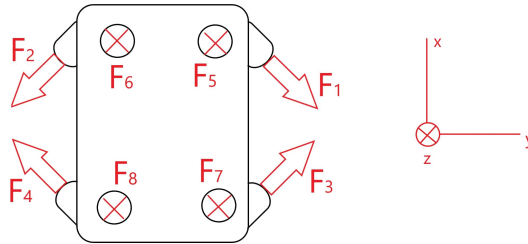
$$\boldsymbol{\tau} = \begin{bmatrix} \mathbf{f} \\ \mathbf{r} \times \mathbf{f} \end{bmatrix} = \begin{bmatrix} F_x \\ F_y \\ F_z \\ F_z l_y - F_y l_z \\ F_x l_z - F_z l_x \\ F_y l_x - F_x l_y \end{bmatrix}. \quad (2.88)$$

Hence, the generalised forces and moments in 6 DoFs  $\boldsymbol{\tau} \in \mathbb{R}^6$  due to 8 thrusters  $\mathbf{F} \in \mathbb{R}^8$  can be then modelled as:

$$\boldsymbol{\tau} = \mathbf{T}(\boldsymbol{\alpha})\mathbf{F} \quad (2.89)$$

where  $\mathbf{T} = [\mathbf{t}_1 \ \mathbf{t}_2 \ \mathbf{t}_3 \ \mathbf{t}_4 \ \mathbf{t}_5 \ \mathbf{t}_6 \ \mathbf{t}_7 \ \mathbf{t}_8]^T \in \mathbb{R}^{6 \times 8}$  is the thrust configuration matrix and  $\boldsymbol{\alpha} \in \mathbb{R}^8$  is the thrust rotation angle vector [20]. As a consequence, the thrust configuration matrix  $\mathbf{T}$  can be then computed by using equation (2.88).

The moment arms of 8 thrusters relative to centre of gravity (CG) are computed and listed in Table 2.3.



**Figure 2.6:** EVA schematic of thrust forces w.r.t. CG

The angles of rotation for the horizontal thrusters from T1 to T4 are as follows:  $\frac{3\pi}{4}$ ,  $-\frac{3\pi}{4}$ ,  $\frac{\pi}{4}$ , and  $-\frac{\pi}{4}$ , respectively. Thrusters T5 to T8, on the other hand, are

$T_i$	$l_{xi}$ (mm)	$l_{yi}$ (mm)	$l_{zi}$ (mm)
$T_1$	143.63	143.63	31.55
$T_2$	152.73	-152.73	31.55
$T_3$	-187.27	187.27	31.55
$T_4$	-196.37	-196.37	31.55
$T_5$	178.86	125.9	0
$T_6$	178.86	-144.1	0
$T_7$	-266.14	125.9	0
$T_8$	-266.14	-144.1	0

**Table 2.3:** Moment arms of 8 thrusters of EVA

vertical thrusters that do not involve any horizontal rotations. Consequently, the forces and moments generated by thruster T1 can be determined.

$$\boldsymbol{\tau}_1 = \begin{bmatrix} F_{x1} \\ F_{y1} \\ F_{z1} \\ F_{z1}l_{y1} - F_{y1}l_{z1} \\ F_{x1}l_{z1} - F_{z1}l_{x1} \\ F_{y1}l_{x1} - F_{x1}l_{y1} \end{bmatrix} = \begin{bmatrix} -F_1 \cos\left(\frac{\pi}{4}\right) \\ F_1 \sin\left(\frac{\pi}{4}\right) \\ 0 \\ -F_1 \sin\left(\frac{\pi}{4}\right) \times 0.032 \\ -F_1 \cos\left(\frac{\pi}{4}\right) \times 0.032 \\ F_1 \sin\left(\frac{\pi}{4}\right) \times 0.144 + F_1 \cos\left(\frac{\pi}{4}\right) \times 0.144 \end{bmatrix}. \quad (2.90)$$

Hence,

$$\boldsymbol{\tau}_1 = F_1 \begin{bmatrix} -0.7071 \\ 0.7071 \\ 0 \\ -0.0223 \\ -0.0223 \\ 0.2032 \end{bmatrix} = F_1 \mathbf{t}_1. \quad (2.91)$$

where we have considered the schematic depicted in Figure 2.6.

By following the same procedure, the forces and moments produced by total 8 thrusters are found to be:

$$\begin{aligned}
 \boldsymbol{\tau} &= \mathbf{T}(\boldsymbol{\alpha})\mathbf{F} = \\
 &= \begin{bmatrix} -0.7071 & -0.7071 & 0.7071 & 0.7071 & 0 & 0 & 0 & 0 \\ 0.7071 & -0.7071 & 0.7071 & -0.7071 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ -0.0223 & 0.0223 & -0.0223 & 0.0223 & 0.1259 & -0.1441 & 0.1259 & -0.1441 \\ -0.0223 & -0.0223 & 0.0223 & 0.0223 & -0.1788 & -0.1788 & 0.2661 & 0.2661 \\ 0.2032 & -0.2160 & -0.2648 & 0.2778 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \\ F_5 \\ F_6 \\ F_7 \\ F_8 \end{bmatrix} \\
 & \quad (2.92)
 \end{aligned}$$

Therefore, the thrust configuration matrix  $\mathbf{T}$  for EVA is given by:

$$\mathbf{T} = \begin{bmatrix} -0.7071 & -0.7071 & 0.7071 & 0.7071 & 0 & 0 & 0 & 0 \\ 0.7071 & -0.7071 & 0.7071 & -0.7071 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ -0.0223 & 0.0223 & -0.0223 & 0.0223 & 0.1259 & -0.1441 & 0.1259 & -0.1441 \\ -0.0223 & -0.0223 & 0.0223 & 0.0223 & -0.1788 & -0.1788 & 0.2661 & 0.2661 \\ 0.2032 & -0.2160 & -0.2648 & 0.2778 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (2.93)$$

The control allocation problem is the computation the control input signal  $\mathbf{u}$  to apply to the thrusters such that the overall desired control forces  $\boldsymbol{\tau}$  can be generalised. First of all, we need to find a relationship between the input signal  $\mathbf{u}$  and the thrust forces  $\mathbf{F}$ . Since EVA uses BlueRobotics T200 Thrusters both for the horizontal and vertical thrusts, we can exploit the following relationship [15]:

We can now derive the inverse of Equation (2.89) as

$$\mathbf{F} = \mathbf{T}^{-1}\boldsymbol{\tau}. \quad (2.94)$$

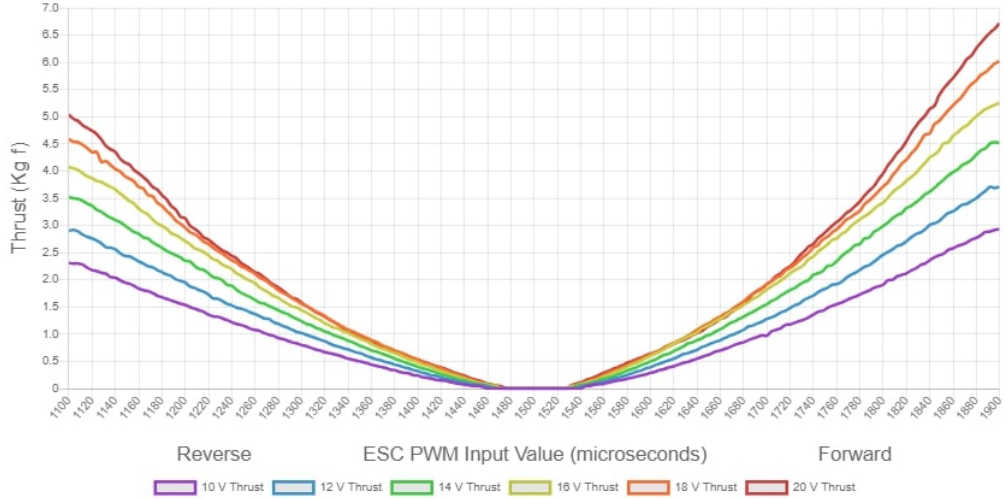
However, since the thrust configuration matrix  $\mathbf{T}$  for EVA is non-square, the Moore-Penrose pseudo-inverse  $\mathbf{T}^+$  is applied given by:

$$\mathbf{T}^+ = \mathbf{T}^T (\mathbf{T}\mathbf{T}^T)^{-1}. \quad (2.95)$$

Hence, the thrust forces vector  $\mathbf{F}$  can be calculated as:

$$\mathbf{F} = \mathbf{T}^+ \boldsymbol{\tau}. \quad (2.96)$$

In this way we can give as input the forces and moments vector  $\boldsymbol{\tau}$  and get as output the relative thrust forces vector  $\mathbf{F}$ . Finally, we can apply to each motor a PWM signal as shown in Figure 2.7 depending on the value of  $\mathbf{F}$  [20].



**Figure 2.7:** T200 BlueRobotics Thrust (kgf) with respect to the ESC PWM Input Value ( $\mu s$ )

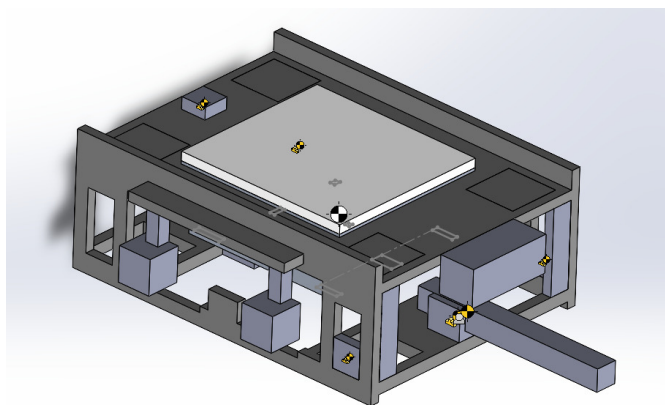
## 2.2.2 Hydrodynamic parameters

In literature, there are several studies that report the work done for calculating the hydrodynamic parameters of ROVs and AUVs. Some estimate the parameters based on carefully collected data, while others use CFD [6][21]. In this subsection, we are gonna discuss the resolution of the problem and the techniques used by the hydrodynamic division of the PoliTOcean team.

In order to calculate the added mass and damping, we performed Dynamic Fluid Body Interaction (DFBI) simulations using the computational fluid dynamics software Star CCM+. DFBI simulations employ a different approach compared to more common simulations. In fact, DFBI simulations use the *overset mesh* technique to allow a body (in this case, the ROV) to move within a domain. The alternative to this simulation is to have a stationary body in the domain subjected to a flow, which does not accurately reproduce the real physics where the water is stationary, and the ROV has its own velocity.

In short, DFBI simulations are carried out by generating two meshes: the first for the domain within which the body will move and a smaller one containing the moving body. These two meshes are overlapped, hence the name *overset mesh*. At each *time-step*, the body will move within the domain. To avoid generating a too large domain, which would lead to high computational costs, the outer domain mesh can also be set in motion by assigning it the same velocity as the moving body. In this way, the relative motion between the reference frame of the body and the external domain will be null, but since the fluid is stationary with respect to both reference frames, it will still be possible to simulate the motion of the ROV in the fluid.

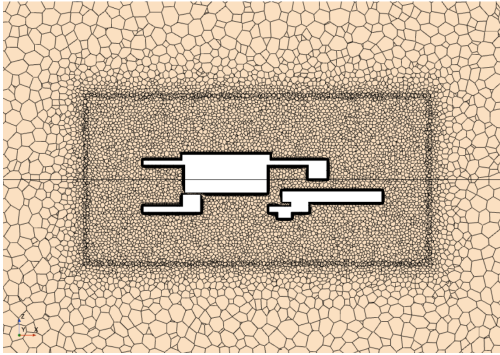
To achieve this, we used a simplified geometry of the ROV that allowed us to maintain a relatively low computational cost.



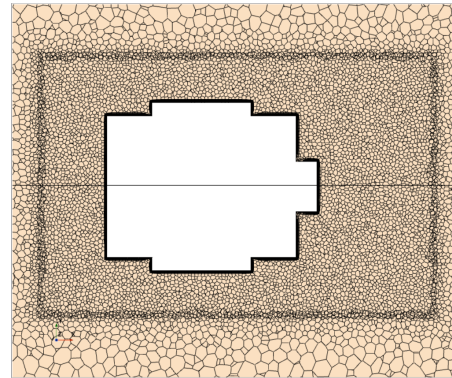
**Figure 2.8:** Simplified CAD of EVA

The type of mesh was chosen to account for the turbulence generated by the squat geometry of the ROV. Therefore, a *polyhedral mesh* was chosen, as the cells generated by this mesh are polyhedrons usually consisting of more than six faces. This is important because in numerical computation, where finite volumes are used (as in Star CCM+), the transmission of information between one cell and another generates a certain error depending on the angle of incidence between the information and the surface. The closer this angle is to a right angle, the smaller the error generated. In areas where turbulence is present, the information comes from every direction, and to obtain an error that does not negatively affect the result, it is important to have cells that are not only small but also have a large number of faces. Another feature of the mesh is the *prism layers* near the body. These were used because it is known that the gradient at the wall is vertical.





**Figure 2.9:** Mesh of the ROV, side section view



**Figure 2.10:** Mesh of the ROV, top section view

We chose to use the *Standard  $k - \varepsilon$  all  $y+$*  turbulence solver, belonging to the family of *Unsteady Reynolds-Averaged Navier-Stokes* (URANS). The reason why the simulations were conducted in an unsteady regime instead of a steady state is twofold: the DFBI method implies that the movement of the ROV varies over time, and to observe the added mass, it was necessary to vary the body's velocity in the domain. We chose to use the *Standard  $k - \varepsilon$  all  $y+$*  solver because we found that it yields results more similar to experimental ones. We chose to use an *all  $y+$*  approach because, given the squat geometry, the drag coefficient is largely generated by pressure and only minimally by shear stress. This means that we can afford to lose some information at the wall without it negatively affecting our results.

## 2.3 Control-oriented model

In this section, mathematical models used for computing control laws will be presented. These models are all linearized and comes from certain simplifying assumptions about the dynamics of the ROV under control. They draw inspiration from lateral and longitudinal models presented in [4] and other works employing similar models. For example in [10] a linearized model is used for heave control design, while in [6] and [8] similar models are used as basic structure for System Identification purposes. However, the key difference lies in the fact that in the cited models, the ROV is assumed to be neutrally buoyant, whereas in those to be presented, the weight force and buoyancy will be taken into account as EVA is positively buoyant.

According to [4] the 6-DOF equations of motion of a ROV can be divided into two non-interacting (or lightly interacting) subsystems:

- Longitudinal subsystem: states  $u$ ,  $w$ ,  $q$ , and  $\theta$ .
- Lateral subsystem: states  $v$ ,  $p$ ,  $r$ ,  $\phi$ , and  $\psi$ .

Starting from these two MIMO models, it is possible to isolate the dynamics and derive SISO models for each dynamic. This is the procedure used below for the derivation of the control oriented models used for the design of the control system of EVA.

Longitudinal subsystem is based on the following assumptions:

- States  $v$ ,  $p$ ,  $r$ ,  $\phi$  are small.
- Higher-order damping can be neglected.
- Diagonal  $\mathbf{M}_A$ .
- Coriolis is simplified assuming  $v$ ,  $w$ ,  $p$ ,  $q$ ,  $r$  are small.
- $x_g = x_b$ .

By adding the assumption  $u = \text{constant}$  we can remove the forward speed state. From the obtained Longitudinal subsystem, we derived the heave model and pitch control oriented models presented in subsections 2.3.1 and 2.3.2.

As made for longitudinal model, the lateral model assumptions are listed below:

- States  $u$ ,  $w$ ,  $p$ ,  $r$ ,  $\phi$ ,  $\theta$  are small.
- Higher-order damping can be neglected.
- Diagonal  $\mathbf{M}_A$ .

- Coriolis is simplified.
- $x_g = x_b$  and  $y_g = y_b$ .

From the obtained Lateral subsystem, we derived the roll control oriented models presented in subsections 2.3.3.

### 2.3.1 Heave model

Following all the assumption of the longitudinal subsystem we can extract the heave model:

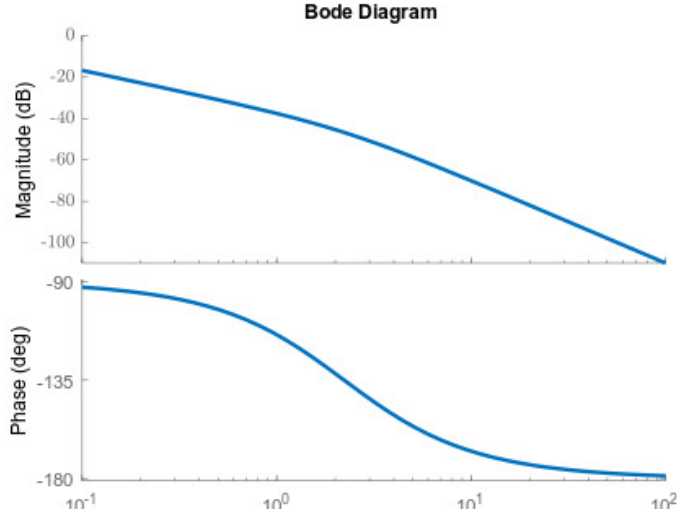
$$(m + |Z_{\dot{w}}|)\dot{w} + |Z_w|w + B - W = \tau_z. \quad (2.97)$$

In state space form, considering the states  $[ w ; z ]$ , becomes:

$$\dot{x} = \begin{bmatrix} \dot{w} \\ w \end{bmatrix} = \begin{bmatrix} \frac{-|Z_w|}{m+|Z_{\dot{w}}|} & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} w \\ z \end{bmatrix} + \begin{bmatrix} \frac{1}{m+|Z_{\dot{w}}|} \\ 0 \end{bmatrix} (\tau_z + W - B) \quad (2.98)$$

$$y = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} w \\ z \end{bmatrix}. \quad (2.99)$$

The bode diagram of the system is reported in 2.11.



**Figure 2.11:** Bode Diagram of Heave model

By discretizing with the Zero Order Hold method the model with sampling time 0.01 seconds, considering  $(\tau_z + W - B)$  as input and  $z$  as output, and substituting the system parameters we obtain the transfer function of the heave dynamics:

$$G_{heave}(z) = \frac{1.6 \times 10^{-6}(z + 0.9926)}{(z - 1)(z - 0.9781)} \quad (2.100)$$

As we can see from 2.100 the system is stable since all the poles are inside the unitary circle in the  $z$  plane. In addition we can notice the presence of a pole located in  $z=1$  that, in closed loop, ensures steady-state zero error for step reference and zero steady-state output error for step disturbances on measurements. Furthermore, as evidenced by the Bode plot and the transfer function, the system is overdamped.

### 2.3.2 Pitch model

Following all the assumptions of the longitudinal subsystem we can extract the pitch model:

$$(I_y + |M_{\dot{q}}|)\ddot{\theta} + |M_q|\dot{\theta} + (z_g W - z_b B) = \tau_{pitch}. \quad (2.101)$$

In state space form, considering the states  $[\ddot{\theta}; \dot{\theta}]$ , becomes:

$$\dot{x} = \begin{bmatrix} \ddot{\theta} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{-|M_q|}{I_y + |M_{\dot{q}}|} & \frac{z_b B}{I_y + |M_{\dot{q}}|} \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta} \\ \theta \end{bmatrix} + \begin{bmatrix} \frac{1}{I_y + |M_{\dot{q}}|} \\ 0 \end{bmatrix} \tau_{pitch} \quad (2.102)$$

$$y = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\theta} \\ \theta \end{bmatrix}. \quad (2.103)$$

The bode diagram of the system is reported in 2.12.

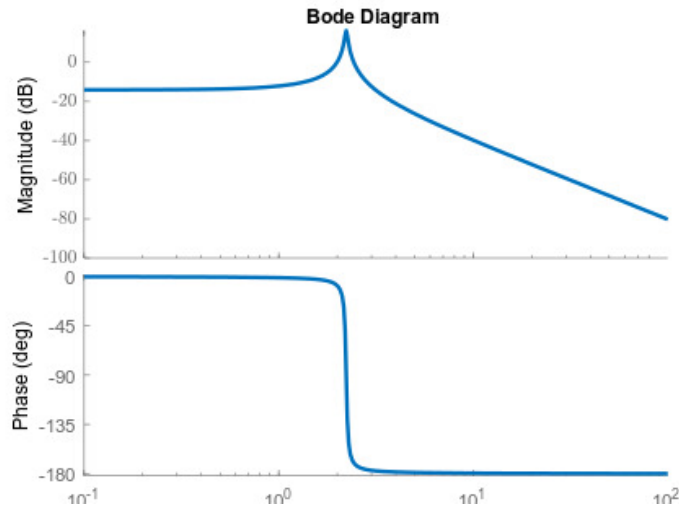


Figure 2.12: Bode Diagram of Pitch model

By discretizing with the Zero Order Hold method the model with sampling time 0.01 seconds, considering  $\tau_{pitch}$  as input and  $\theta$  as output, and substituting the system parameters we obtain the transfer function of the pitch dynamics:

$$G_{pitch}(z) = \frac{4.88 \times 10^{-5}(z + 1)}{z^2 - 1.999z + 0.9993} \quad (2.104)$$

As we can see from 2.104 the system is stable since all the poles are inside the unitary circle in the  $z$  plane. Furthermore, as evidenced by the Bode plot and the transfer function, the system is underdamped.

### 2.3.3 Roll model

Following all the assumptions of the lateral subsystem we can extract the roll model:

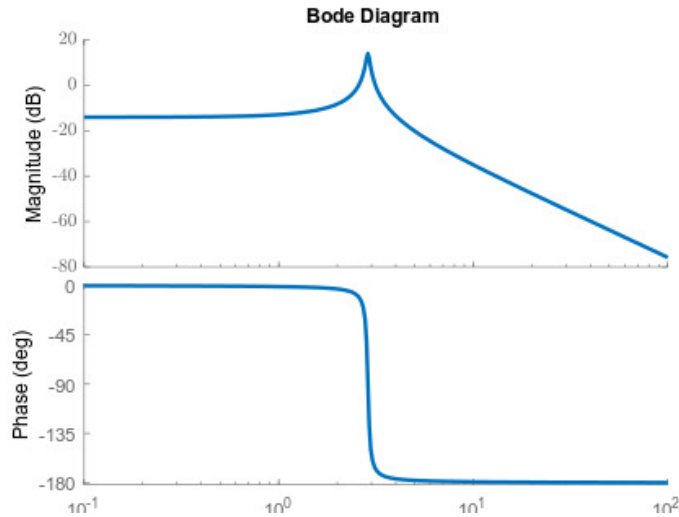
$$(I_x + |K_{\dot{p}}|)\ddot{\phi} + |K_p|\dot{\phi} + (z_g W - z_b B) = \tau_{roll}. \quad (2.105)$$

In state space form, considering the states  $[\ddot{\phi}; \dot{\phi}]$ , becomes:

$$\dot{x} = \begin{bmatrix} \ddot{\phi} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \frac{-|K_p|}{I_x + |K_{\dot{p}}|} & \frac{z_b B}{I_x + |K_{\dot{p}}|} \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \phi \end{bmatrix} + \begin{bmatrix} \frac{1}{I_x + |K_{\dot{p}}|} \\ 0 \end{bmatrix} \tau_{roll} \quad (2.106)$$

$$y = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \phi \end{bmatrix}. \quad (2.107)$$

The bode diagram of the system is reported in 2.13.



**Figure 2.13:** Bode Diagram of Roll model

By discretizing with the Zero Order Hold method the model with sampling time 0.01 seconds, considering  $\tau_{roll}$  as input and  $\phi$  as output, and substituting the system parameters we obtain the transfer function of the pitch dynamics:

$$G_{roll}(z) = \frac{8.1512 \times 10^{-5}(z + 1)}{z^2 - 1.998z + 0.9989} \quad (2.108)$$

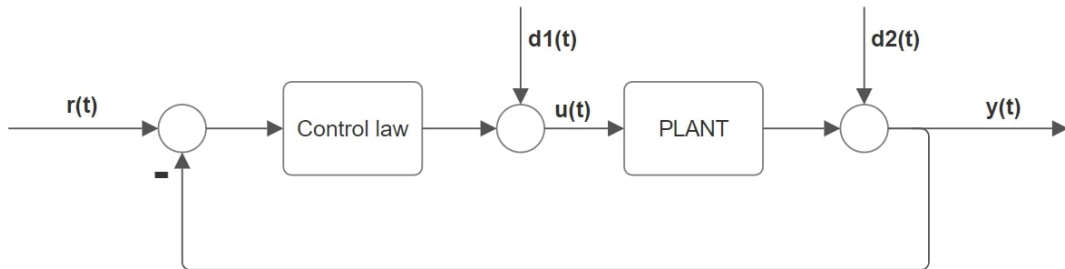
As we can see from 2.108 the system is stable since all the poles are inside the unitary circle in the  $z$  plane. Furthermore, as evidenced by the Bode plot and the transfer function, the system is underdamped.

## Chapter 3

# Control System Design

In this chapter, we will present the various mathematical steps for designing control systems for the ROV. First, a controller will be constructed using pole placement, then control techniques such as LQR and MPC will be explored. All the calculus and performances consideration are based on different models presented in chapter 2 in section 2.3. The comparison of performances on the simulation model of different control system developed in this chapter, will be discussed in chapter 4.

Let's consider the basic 1 DOF controller closed loop structure reported in figure 3.1.



**Figure 3.1:** 1 DOF architecture

The purpose of control for heave dynamics is to meet the following requirements:

- Overshoot  $\hat{s} \leq 10\%$  for step reference signal
- Settling time to 1% of the reference  $t_{s,1\%} \leq 4$  s for step reference signal
- Regime output response  $|y_{d_1}^\infty| = 0$  in the presence of a step disturbance signal  $d_1$

- Regime output response  $|y_{d_2}^\infty| = 0$  in the presence of a step disturbance signal  $d_2$
- Maximum of the control input must be  $\max u(t) \leq 80$  N
- Minimum of the control input must be  $\min u(t) \geq -60$  N.

So the heave controller must be able to track a step signal and to reject different step disturbances that comes from manipulated objects. For what concern pitch and roll dynamics we want that the output of the feedback scheme is always zero; we want a very fast disturbance compensation. In particular the compensation of step disturbances is crucial because the ROV is called to do lot of tasks that involves manipulations. Each time EVA catch an object, from the control system point of view, is like a step disturbance on heave, pitch and roll. The disturbance is mainly on pitch due to the geometry of the ROV, but we cannot assume a priori the shape and the weight distribution of the object under manipulation; depending on what has been said, the external object can also introduce disturbances on the roll dynamic. Furthermore, is crucial to highlight the fact that the system is not perfectly balanced, so compensating for disturbances in all the considered dynamics is necessary to offset all the system unbalances. This is true for very small unbalances, as they introduce non linearities that cannot be represented as external disturbances. So the requirement for roll and pitch control are the following:

- Regime output response  $|y_{d_1}^\infty| = 0$  in the presence of a step disturbance signal  $d_1$
- Regime output response  $|y_{d_2}^\infty| = 0$  in the presence of a step disturbance signal  $d_2$
- Maximum of the control input must be  $\max u(t) \leq 30$  Nm
- Minimum of the control input must be  $\min u(t) \geq -30$  Nm.

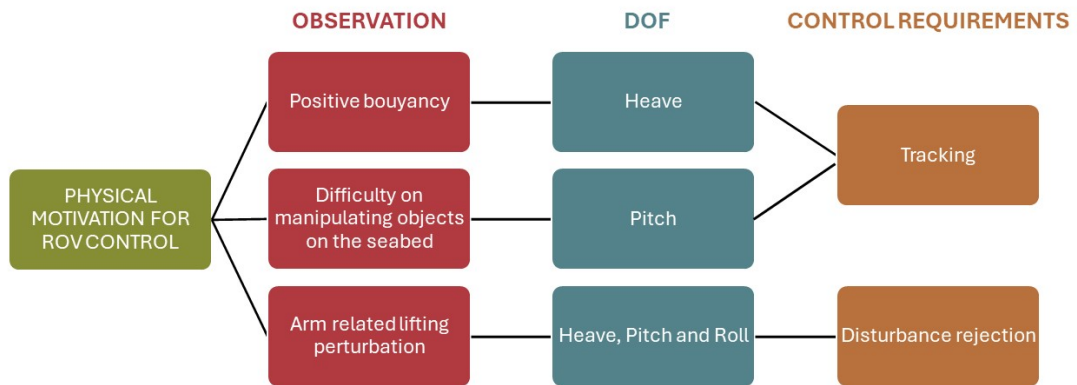
For roll and pitch dynamics, the performance comparison will be based on the system's response speed for disturbance compensation under equal applied disturbances. If the system reacts quickly enough to disturbances, the magnitude of the output displacement due to disturbance  $d_1$  will be lesser, as well as the time required to return to the reference. Similarly, for disturbance  $d_2$ , a better response will allow for faster return to the reference. This is why it is necessary to specify additional requirements for the control design:

- Maximum value of the output due to  $d_1$  must be  $\max y_{d_1} \leq 0.1$  rad
- Settling time to 1% of the reference  $t_{s,1\%}^{d_1} \leq 1$  s.



- Settling time to 1% of the reference  $t_{s,1\%}^{d_2} \leq 1$  s.

Lastly, the ROV has a manipulator with only two degrees of freedom, namely wrist and gripper. Consequently, due to its limited mobility and positioning within the ROV, it may encounter significant difficulties when tasked with lifting small objects resting on the seabed. This is why we want the ROV to be able to adjust its attitude, particularly the pitch angle, to facilitate such manipulations. Consequently, the pitch dynamics must also be controlled for tracking a specific angle not exceeding  $30^\circ$ . The process of identifying issues to address through control can be summarized in the graph shown in figure 3.2.



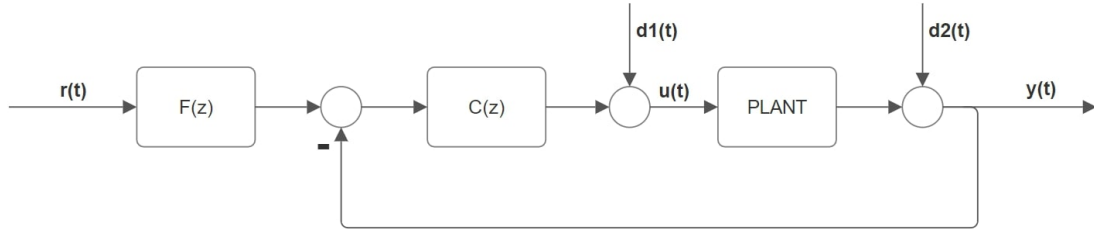
**Figure 3.2:** Control problem identification

In this chapter the design phase on control oriented models is considered. So, the input saturation constraint requirements are omitted. We will deal only with other requirements. We will add anti wind up scheme for thrust saturation (explained in subsection 3.1.1) only in simulation models in chapter 4 where the thruster dynamics is simulated and makes sense to take this phenomena into account.

### 3.1 Control design via pole placement

In this section we are going to discuss all the mathematical procedure for the pole placement control system design. For heave and pitch dynamics, since deal with tracking, we will develop a 2 DOF control system scheme to obtain better

performances, while for roll a 1 DOF scheme is enough since the reference signal is always zero. Every time a 1 DOF control system is discussed, reference will be made to the diagram shown in the figure 3.1, while for what concern 2 DOF control system, the figure 3.3 is considered.



**Figure 3.3:** 2 DOF architecture

The first degree of freedom is the transfer function  $C(z)$ , while the second is the transfer function  $F(z)$ . For the moment, we will consider all the requirements of the problem, except those related to input saturation. In fact, we will address the issue in a dedicated section (subsection 3.1.1) where an anti-windup technique will be presented for dealing with constraints. Once all the mathematical steps for the controller design and the introduction of anti-wind up have been presented, we will examine in detail the results obtained for each of the dynamics that need to be controlled in subsections 3.1.2 3.1.3 3.1.4.

The first step of pole placement control design is checking some basic assumption. Defining:

$$G(z) = \frac{B(z)}{A(z)} \quad C(z) = \frac{S(z)}{R(z)} \quad (3.1)$$

we need to have  $A(z)$  coprime and monic,  $B(z)$  coprime with no roots at  $z = 1$  and  $G(z)$  strictly proper. The closed loop transfer function considering the 1 DOF structure will be:

$$W(z) = \frac{B(z)S(z)}{A(z)R(z) + B(z)S(z)} = \frac{B_m(z)}{A_m(z)}. \quad (3.2)$$

So we can simply design a controller  $C(z)$  able to place desired poles to the closed loop transfer function  $W(z)$  by solving:

$$A(z)R(z) + B(z)S(z) = A_m(z). \quad (3.3)$$

Solving equation 3.3 means computing the unknown polynomials  $R(z)$  and  $S(z)$ . The equation 3.3 is known as Diophantine equation and it admits a solution if the assumption mentioned before are verified.

Due to the fact that we want disturbance rejection for step disturbances we want to introduce a pole in  $z = 1$  in the controller transfer function  $C(z)$ .

We may also want to simplify the closed loop dynamic by introducing stable zero-pole cancellations between  $C(z)$  and  $G(z)$ . In general we cancel factors that correspond to asymptotically stable modes, but in practice is better to follow a rule of thumb based dumping properties. We introduce the cardioid region that is a region of the  $z$  plane that corresponds to roots with higher dumping with respect to the desired closed loop poles.

Taking into account the considerations done till now we can rewrite the problem in a new way. We can rewrite  $G(z)$  as:

$$G(z) = \frac{B(z)}{A(z)} = \frac{B^+(z)B^-(z)}{(z-1)^{l_1}A^+(z)A^-(z)} \quad (3.4)$$

Where  $B^+(z)$  and  $A^+(z)$  are the stable factors inside the cardioid region of  $B(z)$  and  $A(z)$ , while  $B^-(z)$  and  $A^-(z)$  are the remaining factors. The  $l_1$  poles in  $z = 1$  if they exist, are kept separate. To cancel  $B^+(z)$  and  $A^+(z)$  and to impose disturbance rejection for step disturbances the controller must have the following form:

$$C(z) = \frac{S(z)}{R(z)} = \frac{A^+(z)S^*(z)}{(z-1)^{l_2}B^+(z)R^*(z)} \quad (3.5)$$

where  $l_2 = 1$  for our purposes.

The closed loop transfer function, if  $l = l_1 + l_2$ , become:

$$W(z) = \frac{B^-(z)S^*(z)}{(z-1)^l A^-(z)R^*(z) + B^-(z)S^*(z)} = \frac{B_m(z)}{A_m(z)} \quad (3.6)$$

and the Diophantine equation:

$$(z-1)^l A^-(z)R^*(z) + B^-(z)S^*(z) = A_m(z). \quad (3.7)$$

To obtain a unique solution from equation 3.7 we need the number of unknowns equal to the number of algebraic equations. Based on this we can write the following equations for the degree of the involved polynomials:

$$\deg(S^*) = l + \deg(A^-(z)) - 1 \quad (3.8)$$

$$\deg(R^*) = \deg(A^-(z)) + \deg(A^+(z)) + l_1 - \deg(B^+(z)) - 1 \quad (3.9)$$

$$\deg(A_m) = l + \deg(A^-(z)) + \deg(A^+(z)) + l_1 - \deg(B^+(z)) - 1 \quad (3.10)$$

The pole placement is done by imposing the roots of  $A_m$ . Typically, two poles are set based on the requirements using formulas of a  $2^{nd}$  order prototype system.

$$p_{1,2} = -\zeta\omega_n \pm j\omega_n\sqrt{1-\zeta^2} \quad (3.11)$$

where  $\zeta$  and  $\omega_n$  are calculated from requirements with the formulas:

$$\zeta = \frac{|\ln(\hat{s})|}{\sqrt{\pi^2 + \ln^2(\hat{s})}} \quad (3.12)$$

$$\omega_n = \frac{4.6}{\zeta t_{s,1\%}} \quad (3.13)$$

If  $Am$  has a degree greater than two, then the remaining poles are set to be very fast so that the dominant dynamics of the second order is left unchanged.

Till now we referred to the 1 DOF control structure. To realize the 2 DOF structure we need to design the  $F(z)$  transfer function.  $F(z)$  has to be proper and stable to guarantee closed loop stability with all the structure defined before. The denominator is usually designed to cancel slow zeros introduced by  $S^*(z)$  to avoid to affect closed loop response:

$$F(z) = \frac{T(z)}{S^*(z)} \quad (3.14)$$

$$W(z) = F(z) \frac{B^-(z)S^*(z)}{(z-1)^l A^-(z)R^*(z) + B^-(z)S^*(z)} = \quad (3.15)$$

$$= \frac{T(z)B^-(z)}{(z-1)^l A^-(z)R^*(z) + B^-(z)S^*(z)} \quad (3.16)$$

The numerator of  $F(z)$  is designed to place additional zeros in  $W(z)$  to improve output performances and to impose a suitable DC gain on the total closed loop. In this thesis  $T(z)$  is used to cancel the additional faster poles of  $W(z)$  added to the two dominant poles in the 1 DOF phase for obtaining a unique solution from the Diophantine equation. We can also impose a desired DC gain  $k_w$  to  $W(z)$  thanks to  $F(z)$  by writing it as:

$$F_k(z) = k_f \frac{T(z)}{S^*(z)}. \quad (3.17)$$

By calculating the DC gain of the total closed loop transfer function:

$$W(z) = F_k(z) \frac{B^-(z)S^*(z)}{(z-1)^l A^-(z)R^*(z) + B^-(z)S^*(z)} = \quad (3.18)$$

$$= \frac{k_f T(z)B^-(z)}{(z-1)^l A^-(z)R^*(z) + B^-(z)S^*(z)} \quad (3.19)$$

it's easy to see that we can find a suitable  $k_f$  fixing  $k_w$  with the formula:

$$k_f = k_w \frac{S^*(1)}{T(1)} \frac{(z-1)^l A^-(z)R^*(z) + B^-(z)S^*(z)}{B^-(z)S^*(z)}. \quad (3.20)$$

In this way, from the reference tracking point of view, the closed loop behaviour is exactly the  $2^{nd}$  order prototype desired ones[22][23][24].

### 3.1.1 Anti wind-up for saturation handling

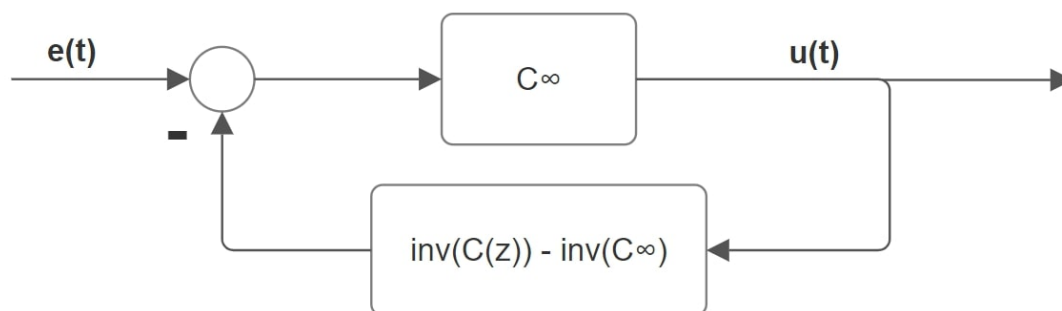
Till now the requirements taken into account in the design are  $\hat{s}$ ,  $t_{s,1\%}$  and  $|y_d^\infty| = 0$  in the presence of a step disturbance signal  $d_1$  or  $d_2$ . In this section a way to deal with input saturation will be presented. This problem is analyzed in [22] and we will consider it in the design from chapter 4.

Since real actuators have limited authority, it is crucial to consider this aspect, as neglecting it can lead to significant performance degradation. A common issue when an input reaches its saturation limit is that, if the controller includes an integrator, it may continue integrating despite the input constraint. As a result, the integrator's state may escalate to an undesirable level, leading to sub optimal transient performance. This problem is known as wind up.

Numerous alternative approaches exist for safeguarding against wind-up. All these techniques center around ensuring that the states of controller have two crucial attributes reported here from [22]:

- the state of the controller should be driven by the actual constrained plant input
- the states of the controller should have a stable realization when driven by the actual plant input.

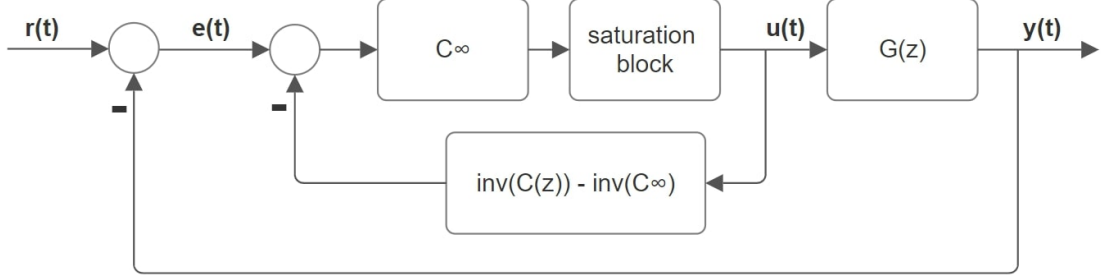
Attaining these conditions is notably straightforward when dealing with a biproper and minimum phase controller. Considering the controller  $C(z)$ , we can decompose it into the direct feedthrough component, denoted as  $C_\infty$ , and a strictly proper transfer function represented by  $\bar{C}(z)$ . If we consider the scheme in figure 3.4 we can see that the transfer function from  $e(t)$  to  $u(t)$  is  $C(z)$ .



**Figure 3.4:** Feedback form of biproper controller

Since  $C(z)$  is minimum phase,  $[C(z)]^{-1}$  is stable so all the bottom part of the scheme is stable and has as input  $u(t)$ . So the listed conditions are satisfied. The

final block scheme that ensure safety from wind up is the same of figure 3.4, but with saturation block placed as in figure 3.5 where all the control loop is represented. In the saturation block we simply saturate the input signal as the requirements on  $u(t)$  impose.



**Figure 3.5:** Anti wind-up control loop

In the design phase on control oriented models the input saturation constraint are not considered. We will deal only with other requirements. We will add anti wind up scheme for thrust saturation only in simulation models where the thruster dynamics is simulated and makes sense to take this phenomena into account.

### 3.1.2 Heave control via pole placement

For depth control, the poles were selected to achieve better performances compared to the requirements, adopting a more conservative approach in anticipation of the degradation present in the simulation model due to non idealities. With  $\hat{s} = 0\%$  and  $t_{s,1\%} = 1$  s, we obtain  $\zeta = 1$  and  $\omega_n = 6.64$  rad/s. Since the heave transfer function is the one in equation 2.100, considering the cardioid region defined by *zeta*, we can define:

$$A^+(z) = (z - 0.9781); \quad A^-(z) = 1 \quad (3.21)$$

and:

$$B^+(z) = 1; \quad B^-(z) = 1.6 \times 10^{-6}(z + 0.9926) \quad (3.22)$$

Using equations 3.8 3.9 3.10 we have:

$$\text{deg}(S^*) = 1; \quad \text{deg}(R^*) = 1; \quad \text{deg}(Am) = 3. \quad (3.23)$$

So we have to place 3 poles, 2 that depends on equation 3.11 and one that is an additional faster pole. This poles in discrete time are:

$$p_{1,2} = 0.9357; \quad p_3 = 0.7175. \quad (3.24)$$

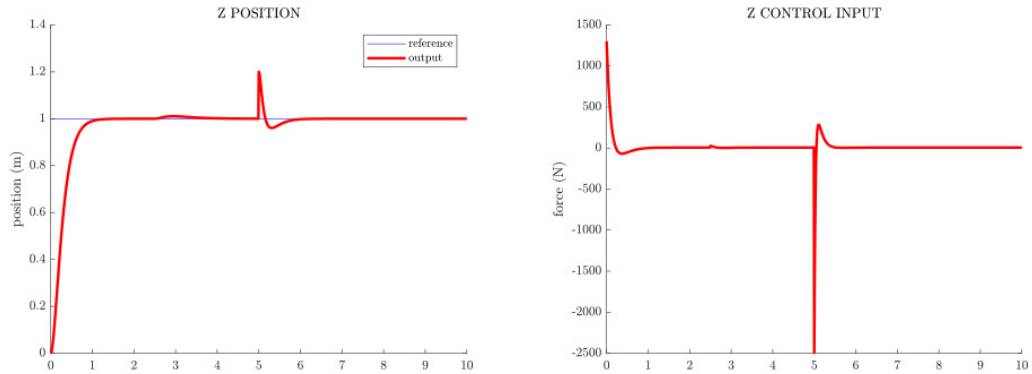
Solving Diophantine equation 3.7, considering that  $C(z)$  must have a pole in  $z = 1$  for the requirements on disturbances, according to equation 3.5, we obtain:

$$C(z) = \frac{A^+(z)S^*(z)}{(z-1)^2 B^+(z)R^*(z)} = \frac{12499(z-0.9781)(z-0.9707)}{(z-1)(z-0.609)}. \quad (3.25)$$

Since the control in z-axis deals with tracking, we can improve performances with a 2 DOF structure. The transfer function  $F(z)$  have the form of equation 3.14. The denominator is equal to  $S^*(z)$  to cancel the slow zeros in the closed loop response, while the numerator is used to cancel the effect of the additional fast poles  $p_3$  of  $W(z)$ . The gain  $k_f$  is calculated thanks to equation 3.20, imposing  $k_w = 1$ . At the end  $F(z)$  is:

$$F(z) = \frac{0.10359(z-0.7175)}{(z-0.9707)}. \quad (3.26)$$

The obtained control law has been tested in MATLAB/Simulink with a 2 DOF structure as in figure 3.3. The reference signal is a step signal of amplitude 1 m. At time  $t = 2.5$  s we introduced a step disturbances  $d_1 = 20$  N, while at time  $t = 5$  s a step disturbances  $d_2 = 0.2$  m. The output response and the control input are plotted in figure 3.6



**Figure 3.6:** Pole placement control for heave dynamic

As we can see the requirements are satisfied with a safety margin since:

$$\hat{s} = 0; \quad t_{s,1\%} = 1 \text{ s}; \quad |y_{d_1, d_2}^\infty| = 0. \quad (3.27)$$

Input saturation is considered in chapter 4 with the anti wind-up techniques explained in subsection 3.1.1.

### 3.1.3 Pitch control via pole placement

For pitch control, the poles were selected to achieve a fast disturbance rejection, adopting a more conservative approach in anticipation of the degradations present

in the simulation model due to non idealities. With  $\hat{s} = 0\%$  and  $t_{s,1\%} = 0.5$  s, we obtain  $\zeta = 1$  and  $\omega_n = 13.28$  rad/s. Since the pitch transfer function is the one in equation 2.104, considering the cardioid region defined by *zeta*, we can define:

$$A^+(z) = 1; \quad A^-(z) = z^2 - 1.999z + 0.9993 \quad (3.28)$$

and:

$$B^+(z) = 1; \quad B^-(z) = 4.8804 \times 10^{-5}(z + 1) \quad (3.29)$$

Using equations 3.8 3.9 3.10 we have:

$$\deg(S^*) = 2; \quad \deg(R^*) = 1; \quad \deg(Am) = 4. \quad (3.30)$$

So we have to place 4 poles, 2 that depends on equation 3.11 and two that are additional faster poles. This poles in discrete time are:

$$p_{1,2} = 0.8756; \quad p_{3,4} = 0.5148. \quad (3.31)$$

Solving Diophantine equation 3.7, considering that  $C(z)$  must have a pole in  $z = 1$  for the requirements on disturbances, according to equation 3.5, we obtain:

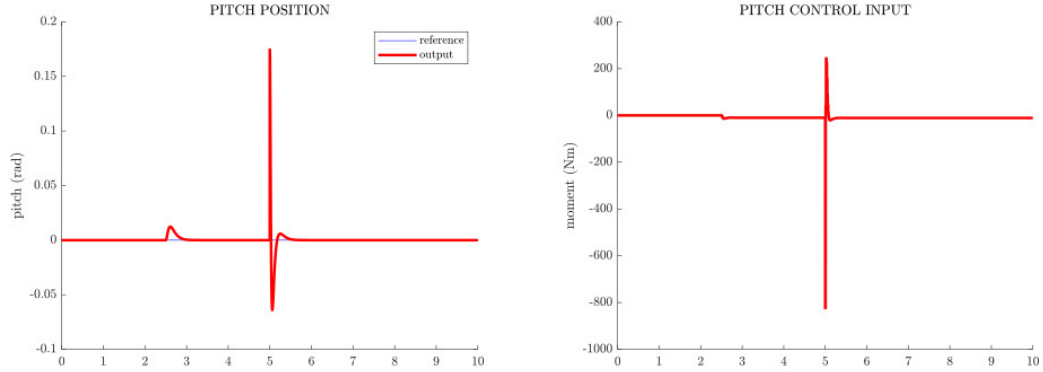
$$C(z) = \frac{A^+(z)S^*(z)}{(z-1)^2 B^+(z)R^*(z)} = \frac{4661.7(z^2 - 1.843z + 0.8514)}{(z-1)(z - 0.009544)}. \quad (3.32)$$

Since the pitch control deals with tracking, we can improve performances with a 2 DOF structure. The transfer function  $F(z)$  have the form of equation 3.14. The denominator is equal to  $S^*(z)$  to cancel the slow zeros in the closed loop response, while the numerator is used to cancel the effect of the additional fast poles  $p_3$  of  $W(z)$ . The gain  $k_f$  is calculated thanks to equation 3.20, imposing  $k_w = 1$ . At the end  $F(z)$  is:

$$F(z) = \frac{0.016493(z - 0.5148)}{(z^2 - 1.843z + 0.8514)}. \quad (3.33)$$

The obtained control law has been tested in MATLAB/Simulink with a 2 DOF structure as in figure 3.3. In the first simulation, the reference signal is a constant signal of amplitude zero. At time  $t = 2.5$  s we introduced a step disturbances  $d_1 = 10$  Nm, while at time  $t = 5$  s a step disturbances  $d_2 = 10$  degree. The output response and the control input are plotted in figure 3.7.



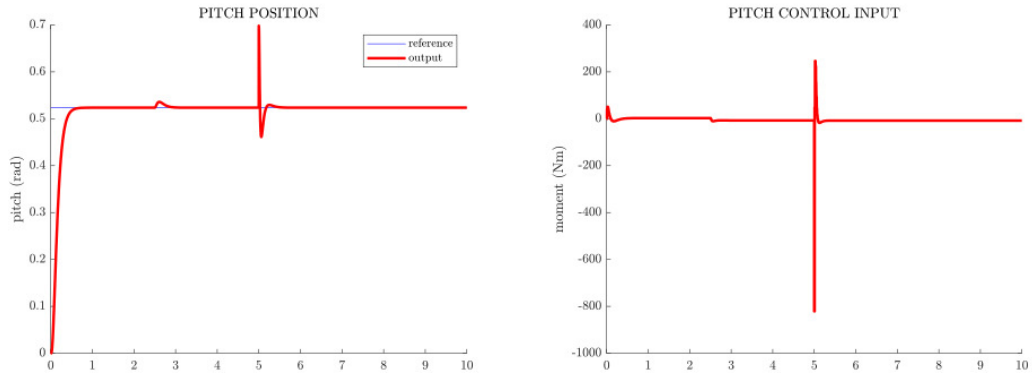


**Figure 3.7:** Pole placement regulation for pitch dynamic

As we can see the requirements are satisfied with a safety margin since:

$$|y_{d_1, d_2}^\infty| = 0; \quad \max y_{d_1} = 0.012 \text{ rad}; \quad t_{s,1\%}^{d_1} = 0.17 \text{ s}; \quad t_{s,1\%}^{d_2} = 0.14 \text{ s}. \quad (3.34)$$

In a second simulation everything is left unchanged except for reference signal that is now an angle of  $\frac{\pi}{6}$  rad. As we can see in the next figures, tracking is achieved.



**Figure 3.8:** Pole placement tracking for pitch dynamic

Input saturation is considered in chapter 4 with the anti wind-up techniques explained in subsection 3.1.1.

### 3.1.4 Roll control via pole placement

For roll control, the poles were selected to achieve a fast disturbance rejection, adopting a more conservative approach in anticipation of the degradation present in the simulation model due to non idealities. With  $\hat{s} = 0\%$  and  $t_{s,1\%} = 0.5$  s, we

obtain  $\zeta = 1$  and  $\omega_n = 13.28$  rad/s. Since the roll transfer function is the one in equation 2.108, considering the cardioid region defined by *zeta*, we can define:

$$A^+(z) = 1; \quad A^-(z) = z^2 - 1.998z + 0.9989 \quad (3.35)$$

and:

$$B^+(z) = 1; \quad B^-(z) = 8.1512 \times 10^{-5}(z + 1) \quad (3.36)$$

Using equations 3.8 3.9 3.10 we have:

$$\deg(S^*) = 2; \quad \deg(R^*) = 1; \quad \deg(Am) = 4. \quad (3.37)$$

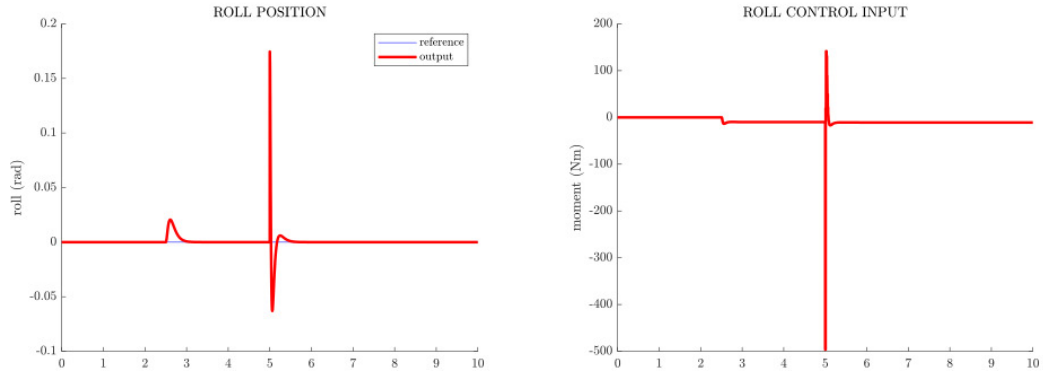
So we have to place 4 poles, 2 that depends on equation 3.11 and two that are additional faster poles. This poles in discrete time are:

$$p_{1,2} = 0.8756; \quad p_{3,4} = 0.5148. \quad (3.38)$$

Solving Diophantine equation 3.7, considering that  $C(z)$  must have a pole in  $z = 1$  for the requirements on disturbances, according to equation 3.5, we obtain:

$$C(z) = \frac{A^+(z)S^*(z)}{(z-1)^2 B^+(z)R^*(z)} = \frac{2785.5(z^2 - 1.844z + 0.8518)}{(z-1)(z-0.009875)}. \quad (3.39)$$

The obtained control law has been tested in MATLAB/Simulink with a 1 DOF structure as in figure 3.1. The reference signal is a constant signal of amplitude zero. At time  $t = 2.5$  s we introduced a step disturbances  $d_1 = 10$  Nm, while at time  $t = 5$  s a step disturbances  $d_2 = 10$  degree. The output response and the control input are plotted in figure 3.9.



**Figure 3.9:** Pole placement control for roll dynamic

As we can see the requirements are satisfied with a safety margin since:

$$|y_{d_1, d_2}^\infty| = 0; \quad \max y_{d_1} = 0.020 \text{ rad}; \quad t_{s,1\%}^{d_1} = 0.24 \text{ s}; \quad t_{s,1\%}^{d_2} = 0.14 \text{ s}. \quad (3.40)$$

Input saturation is considered in chapter 4 with the anti wind-up techniques explained in subsection 3.1.1.

## 3.2 Control design through LQR

In this section we are gonna discuss all the mathematical procedure for Linear Quadratic Regulator (LQR) design. The LQR problem is formulated in lot of papers and books. The ones used for the formulation of our problem are [22][25][26][27].

LQR is an optimal control technique whose main objective is to minimize a quadratic cost function that combines system state and control inputs, suitably weighted during the design phase according to the specifications. LQR control is based on optimal control theory, specifically on the Riccati equation, which is used to compute the optimal control law.

In the following passages the mathematical model of the plant is a state space form. In our applications this are the control-oriented models presented in setion 2.3. Evaluations of control performance on control-oriented models will be conducted in this chapter, as with all other types of control discussed in this thesis. Subsequently, in Chapter 4, the performance of the controllers obtained will be compared on simulation model. The discrete time LQR control for a linear system of order  $n$  is usually based on the following optimization problem:

$$\begin{aligned} \min_{u(k)} \sum_{i=0}^{\infty} \mathbf{x}^T(k-i) \mathbf{Q} \mathbf{x}(k+i) + u^T(k+i) R u(k+i) \\ \text{s.t. } \mathbf{x}(k+1) = \mathbf{A} \mathbf{x}(k) + \mathbf{B} u(k) \end{aligned} \quad (3.41)$$

where  $\mathbf{Q} = \mathbf{Q}^T \geq 0$  and  $R \geq 0$  are design parameters that are chosen according to the desired performance tradeoff. It can be proven that:

$$\text{if } \rho(\mathbf{M}_R(\mathbf{A}, \mathbf{B})) = n \quad \& \quad \rho(\mathbf{M}_O(\mathbf{A}, \mathbf{C}_q)) = n, \quad (3.42)$$

the existence and uniqueness of the solution and asymptotic closed loop stability are guarantee.  $\mathbf{C}_q$  is the Choleski factor of matrix  $\mathbf{Q}$  hence, it holds that:

$$\mathbf{Q} = \mathbf{C}_q^T \mathbf{C}_q. \quad (3.43)$$

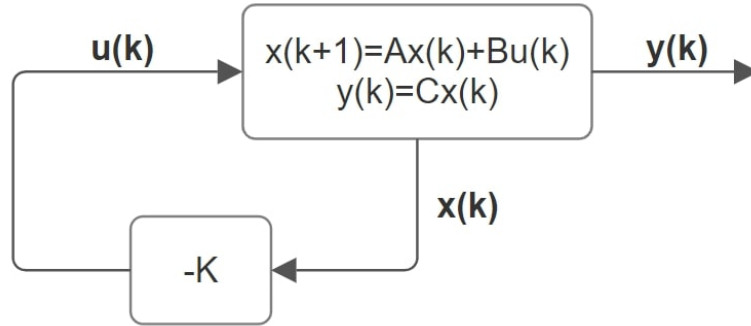
If all the assumptions made above hold true, then the solution to the optimal control problem is given by:

$$u^*(k) = -\mathbf{K} \mathbf{x}(k) \quad \text{with} \quad \mathbf{K} = (\mathbf{R} + \mathbf{B}^T \mathbf{P} \mathbf{B})^{-1} \mathbf{B}^T \mathbf{P} \mathbf{A} \quad (3.44)$$

where  $\mathbf{P} = \mathbf{P}^T \geq 0$  is the solution of the following Discrete Algebraic Riccati Equation:

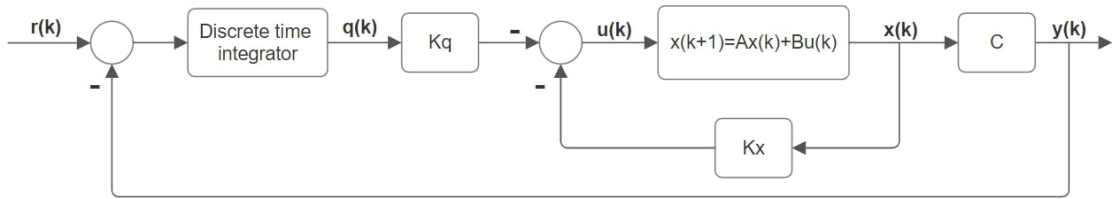
$$\mathbf{P} = \mathbf{A}^T \mathbf{P} \mathbf{A} + \mathbf{Q} - \mathbf{A}^T \mathbf{P} \mathbf{B} (\mathbf{R} + \mathbf{B}^T \mathbf{P} \mathbf{B})^{-1} \mathbf{B}^T \mathbf{P} \mathbf{A}. \quad (3.45)$$

This control law is useful for regulation purpose with the following block scheme:



**Figure 3.10:** Static state feedback control architecture

If our purpose is not regulation, but tracking or step disturbance rejection, this control law leads to weak steady state performances. Hence, we need to modify the control structure and compute a control action that take into account the tracking error as shown below:



**Figure 3.11:** LQR with set point tracking structure

In this structure we introduced the integral  $q(k)$  of the tracking error as an additional state of the system. Hence, calling  $T_s$  the sampling time, we have:

$$\frac{q(k+1) - q(k)}{T_s} = r(k) - y(k) \quad \longrightarrow \quad q(k+1) = q(k) + T_s e(k). \quad (3.46)$$

The coefficient  $K_q$  comes from the modified control law that is obtained repeating the passages explained above, but with the matrices of the augmented system obtained by including in the system state the time integral of the tracking error:

$$\mathbf{x}_{aug}(k) = \begin{bmatrix} q(k) \\ \mathbf{x}(k) \end{bmatrix} \in \mathbb{R}^{n+1}. \quad (3.47)$$

The state equations of the augmented system are now:

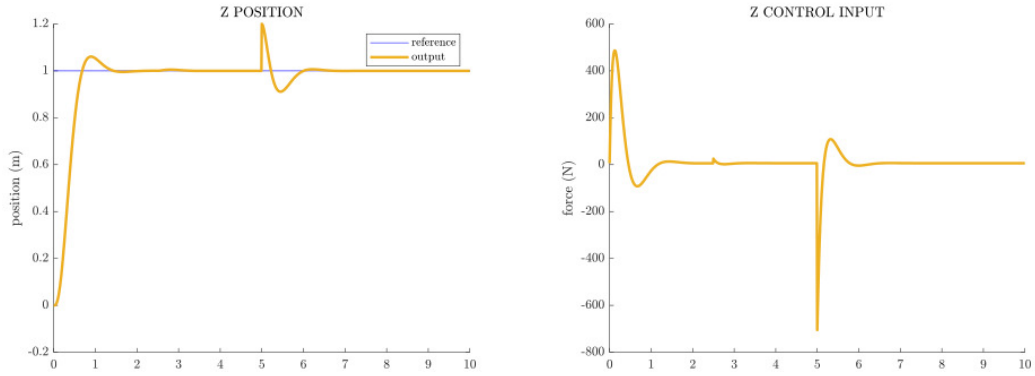
$$\mathbf{x}_{aug}(k+1) = \begin{bmatrix} 1 & -T_s \mathbf{C} \\ \mathbf{0}^{n \times 1} & \mathbf{A} \end{bmatrix} \begin{bmatrix} q(k) \\ \mathbf{x}(k) \end{bmatrix} + \begin{bmatrix} 0 \\ \mathbf{B} \end{bmatrix} u(k) + \begin{bmatrix} T_s \\ \mathbf{0}^{n \times 1} \end{bmatrix} r(k) \quad (3.48)$$

$$y(k) = \begin{bmatrix} 0 & \mathbf{C} \end{bmatrix} \begin{bmatrix} q(k) \\ \mathbf{x}(k) \end{bmatrix}. \quad (3.49)$$

Replacing the new matrix of the system in the passages explained above we will obtain the following modified augmented control law:

$$u^*(k) = -\mathbf{K}_{aug} \mathbf{x}_{aug}(k) = - \begin{bmatrix} K_q & K_x \end{bmatrix} \begin{bmatrix} q(k) \\ \mathbf{x}(k) \end{bmatrix}. \quad (3.50)$$

It can be shown that if the original  $\mathbf{A}$  and  $\mathbf{B}$  matrix of the system are reachable, also the augmented system is fully reachable. Furthermore due to the presence of the integrator in the feedback law, the asymptotic stability is guarantee; the system will have zero steady state tracking error for constant reference and disturbance rejection for step disturbances. The design of LQR for heave, roll and pitch dynamic of EVA is based on the procedure explained before; the design parameters  $\mathbf{Q}$  and  $\mathbf{R}$  were tuned in each controller to fit requirements. The structure used is the one reported in figure 3.11. The result of simulations on control oriented models are in the next figures. For what regard heave, the reference signal is a step signal of amplitude 1 m. At time  $t = 2.5$  s we introduced a step disturbances  $d_1 = 20$  N, while at time  $t = 5$  s a step disturbances  $d_2 = 0.2$  m. The output response and the control input are plotted in figure 3.12.

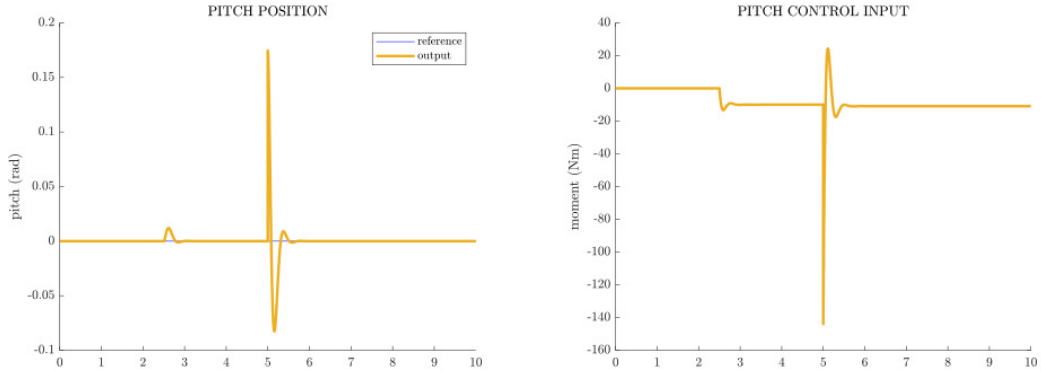


**Figure 3.12:** LQR for heave dynamic

As we can see the requirements are satisfied with a safety margin since:

$$\hat{s} = 0.060 \text{ m}; \quad t_{s,1\%} = 1.30 \text{ s}; \quad |y_{d_1, d_2}^\infty| = 0. \quad (3.51)$$

For pitch control, in a first simulation, the reference signal is a constant signal of amplitude zero. At time  $t = 2.5 \text{ s}$  we introduced a step disturbances  $d_1 = 10 \text{ Nm}$ , while at time  $t = 5 \text{ s}$  a step disturbances  $d_2 = 10 \text{ degree}$ . The output response and the control input are plotted in figure 3.13.

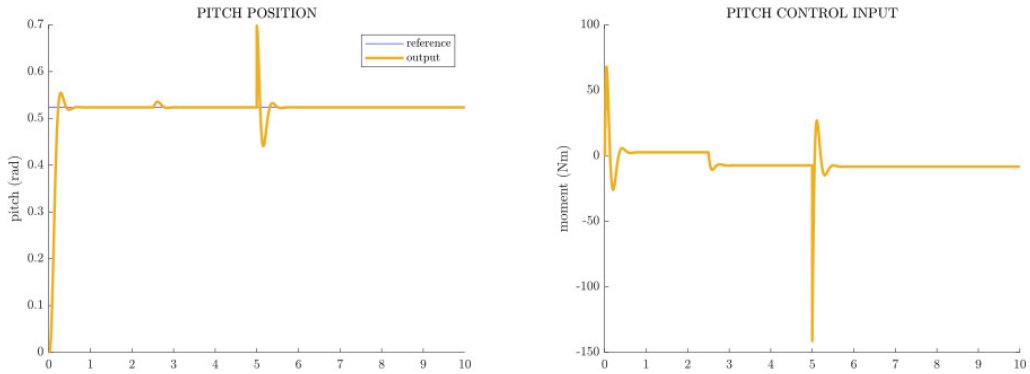


**Figure 3.13:** LQR regulation for pitch dynamic

As we can see the requirements are satisfied with a safety margin since:

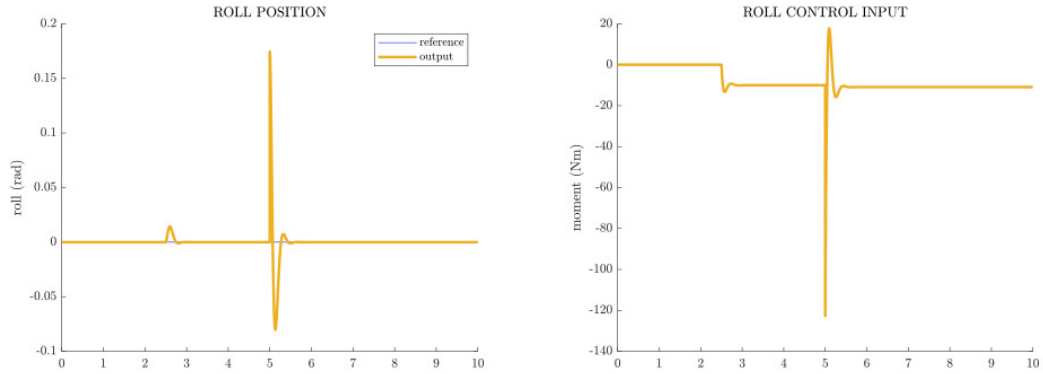
$$|y_{d_1, d_2}^\infty| = 0; \quad \max y_{d_1} = 0.012 \text{ rad}; \quad t_{s, 1\%}^{d_1} = 0.16 \text{ s}; \quad t_{s, 1\%}^{d_2} = 0.28 \text{ s}. \quad (3.52)$$

In a second simulation everything is left unchanged except for reference signal that is now an angle of  $\frac{\pi}{6} \text{ rad}$ . As we can see in the next figures, tracking is achieved.



**Figure 3.14:** LQR tracking for pitch dynamic

For roll control, the reference signal is a constant signal of amplitude zero. At time  $t = 2.5 \text{ s}$  we introduced a step disturbances  $d_1 = 10 \text{ Nm}$ , while at time  $t = 5 \text{ s}$  a step disturbances  $d_2 = 10 \text{ degree}$ . The output response and the control input are plotted in figure 3.15.



**Figure 3.15:** LQR for roll dynamic

As we can see the requirements are satisfied with a safety margin since:

$$|y_{d_1, d_2}^\infty| = 0; \quad \max y_{d_1} = 0.014 \text{ rad}; \quad t_{s,1\%}^{d_1} = 0.15 \text{ s}; \quad t_{s,1\%}^{d_2} = 0.25 \text{ s}. \quad (3.53)$$

Until now, the requirements regarding control input saturation have not been taken into consideration. Similarly to what has been done for the control laws presented in this chapter so far, these requirements will be addressed in chapter 4, where we will compare the performance of all the control systems, on the simulation model. Regarding LQR control, the windup cannot be avoided like for controllers in transfer function form. Consequently, taking into account the input saturation requirements is considered the introduction of a new type of optimal control that solves the constraint problem a priori. The control system in question is Model Predictive Control, discussed in the following section.

### 3.3 MPC design

In this section we are gonna discuss all the mathematical procedure for Model Predictive Control (MPC) design. The MPC problem is formulated in lot of papers and books. The ones used for the formulation of our problem are [28][29]. Unlike traditional control systems that immediately determine control action in response to the current state of the system, MPC makes control decisions predictively by looking ahead in time through a dynamic model of the system. In other words, it forecasts how the system will evolve in the future, considering a defined time window, and selects the optimal control action that minimizes a cost function subject to constraints.

The basic idea of MPC is to use a model of the plant to predict the future evolution of the process to optimize the control signal. The model is of course a simplified model since it has to run in real time and to compute prediction and

optimization at each time step. Of course the big advantage of this kind of control techniques is that we can take into account saturations on inputs, outputs and states, directly on the constraint of the optimization problem involved.

The philosophy of MPC is receding horizon principle: at time  $k$  it solves an optimal control problem over a finite future horizon called prediction horizon  $N_p$ . Then it apply only the first optimal move  $u^*(t)$ , neglecting the rest of the trajectory and at time  $k + 1$ , it get new measurements and repeat the optimization. So the prediction horizon define a moving horizon window that dictates how far we wish to try to predict the future. This process works well thanks to the advantages obtained from the feedback structure, but we have to find a good trade off on the model of the plant involved in optimization. The model has to be descriptive enough to describe the most significant dynamics of the system, but also simple enough for solving the optimization problem each time step in real time.

The objective function to be minimized has to be related to an error function based on the difference between the desired and the actual responses plus a term that express how much control effort we want to consider.

In this thesis the models used for prediction are the linearized control-oriented models. Them are simple enough for online optimization, but also expressive enough. The prove is in chapter 4 where we will see how the control laws designed in this chapter, work well in the ROV simulator. After the explanation of all the mathematical procedure in MPC, we will see performances on control-oriented models, while in chapter 4 we will see how it works on simulation model.

We said that the prediction models are the one introduced in section 2.3, but we want to change them to suit all the following procedure where an integrator is embedded. Calling  $\mathbf{x}_m$  the original states and  $\mathbf{A}_m$ ,  $\mathbf{B}_m$  and  $\mathbf{C}_m$  the original system matrices, the equivalent system with the embedded integrator for a system of order  $n$  is:

$$\begin{bmatrix} \Delta \mathbf{x}_m(k+1) \\ y(k+1) \end{bmatrix} = \begin{bmatrix} \mathbf{A}_m & \mathbf{0}^{n \times 1} \\ \mathbf{C}_m \mathbf{A}_m & 1 \end{bmatrix} \begin{bmatrix} \Delta \mathbf{x}_m(k) \\ y(k) \end{bmatrix} + \begin{bmatrix} \mathbf{B}_m \\ \mathbf{C}_m \mathbf{B}_m \end{bmatrix} \Delta u(k) \quad (3.54)$$

$$y(k) = \begin{bmatrix} \mathbf{0}^{1 \times n} & 1 \end{bmatrix} \begin{bmatrix} \Delta \mathbf{x}_m(k) \\ y(k) \end{bmatrix}. \quad (3.55)$$

So from now on in this section we will refer to these matrices with the following notation:

$$\begin{aligned} \mathbf{x}(k) &= \begin{bmatrix} \Delta \mathbf{x}_m(k) \\ y(k) \end{bmatrix}; & \mathbf{A} &= \begin{bmatrix} \mathbf{A}_m & \mathbf{0}^{n \times 1} \\ \mathbf{C}_m \mathbf{A}_m & 1 \end{bmatrix}; \\ \mathbf{B} &= \begin{bmatrix} \mathbf{B}_m \\ \mathbf{C}_m \mathbf{B}_m \end{bmatrix}; & \mathbf{C} &= \begin{bmatrix} \mathbf{0}^{1 \times n} & 1 \end{bmatrix}. \end{aligned} \quad (3.56)$$



After creating the mathematical model, the next stage in designing a predictive control system, involves predicting the plant's output using the future control signal as adjustable factors. This forecasting is described within the optimization window of length  $N_p$  discussed before. We assume that at time  $k_i$  we have access to the state variable vector  $\mathbf{x}(k_i)$ . This vector,  $\mathbf{x}(k_i)$ , gives us the current status of the system. With given measurement  $\mathbf{x}(k_i)$ , the future state variables are predicted for  $N_p$  number of samples and denoted as:

$$\mathbf{x}(k_i + 1 | k_i), \mathbf{x}(k_i + 2 | k_i), \dots, \mathbf{x}(k_i + m | k_i), \dots, \mathbf{x}(k_i + N_p | k_i) \quad (3.57)$$

where  $\mathbf{x}(k_i + m | k_i)$  is the predicted state at time  $k_i + m$  knowing  $\mathbf{x}(k_i)$ .

The future control actions are denoted as:

$$\Delta u(k_i), \Delta u(k_i + 1), \dots, \Delta u(k_i + N_c - 1) \quad (3.58)$$

where we called  $N_c$  the control horizon. It is usually chosen to be less or equal to the prediction horizon. While the prediction horizon is the time period into the future over which the controller predicts the system's behavior, the control horizon is the time period into the future over which the controller actively computes and implements optimal control actions. It indicates how many time steps the controller considers in the decision-making process to determine optimal control actions. We can calculate the future states sequentially as:

$$\begin{aligned} \mathbf{x}(k_i + 1 | k_i) &= \mathbf{A}\mathbf{x}(k_i) + \mathbf{B}\Delta u(k_i) \\ \mathbf{x}(k_i + 2 | k_i) &= \mathbf{A}\mathbf{x}(k_i + 1 | k_i) + \mathbf{B}\Delta u(k_i + 1) \\ &= \mathbf{A}^2\mathbf{x}(k_i) + \mathbf{A}\mathbf{B}\Delta u(k_i) + \mathbf{B}\Delta u(k_i + 1) \\ &\vdots \\ \mathbf{x}(k_i + N_p | k_i) &= \mathbf{A}^{N_p}\mathbf{x}(k_i) + \mathbf{A}^{N_p-1}\mathbf{B}\Delta u(k_i) \\ &\quad + \mathbf{A}^{N_p-2}\mathbf{B}\Delta u(k_i + 1) + \dots + \mathbf{A}^{N_p-N_c}\mathbf{B}\Delta u(k_i + N_c - 1). \end{aligned} \quad (3.59)$$

Now we can simply substitute to obtain predicted output from the predicted states:

$$\begin{aligned} y(k_i + 1 | k_i) &= \mathbf{C}\mathbf{A}\mathbf{x}(k_i) + \mathbf{C}\mathbf{B}\Delta u(k_i) \\ y(k_i + 2 | k_i) &= \mathbf{C}\mathbf{A}^2\mathbf{x}(k_i) + \mathbf{C}\mathbf{A}\mathbf{B}\Delta u(k_i) + \mathbf{C}\mathbf{B}\Delta u(k_i + 1) \\ y(k_i + 3 | k_i) &= \mathbf{C}\mathbf{A}^3\mathbf{x}(k_i) + \mathbf{C}\mathbf{A}^2\mathbf{B}\Delta u(k_i) + \mathbf{C}\mathbf{A}\mathbf{B}\Delta u(k_i + 1) + \mathbf{C}\mathbf{B}\Delta u(k_i + 2) \\ &\vdots \\ y(k_i + N_p | k_i) &= \mathbf{C}\mathbf{A}^{N_p}\mathbf{x}(k_i) + \mathbf{C}\mathbf{A}^{N_p-1}\mathbf{B}\Delta u(k_i) + \mathbf{C}\mathbf{A}^{N_p-2}\mathbf{B}\Delta u(k_i + 1) + \dots \\ &\quad + \mathbf{C}\mathbf{A}^{N_p-N_c}\mathbf{B}\Delta u(k_i + N_c - 1). \end{aligned} \quad (3.60)$$

Of course since our available information is  $\mathbf{x}(k_i)$  all the predictions are formulated in terms of it and in terms of  $\Delta u(k_i + j)$  with  $j = 0, 1, 2, \dots, N_c - 1$  that is the future control movement. We can rewrite the output prediction given  $\mathbf{x}(k_i)$  and  $\Delta u(k_i + j)$  with  $j = 0, 1, 2, \dots, N_c - 1$  in compact form as:

$$\mathbf{Y} = \mathbf{F}\mathbf{x}(k_i) + \Phi\Delta\mathbf{U} \quad (3.61)$$

where we defined:

$$\begin{aligned} \mathbf{Y} &= [y(k_i + 1 | k_i) y(k_i + 2 | k_i) y(k_i + 3 | k_i) \dots y(k_i + N_p | k_i)]^T; \\ \Delta\mathbf{U} &= [\Delta u(k_i) \Delta u(k_i + 1) \Delta u(k_i + 2) \dots \Delta u(k_i + N_c - 1)]^T; \\ \mathbf{F} &= \begin{bmatrix} \mathbf{CA} \\ \mathbf{CA}^2 \\ \mathbf{CA}^3 \\ \vdots \\ \mathbf{CA}^{N_p} \end{bmatrix}; \Phi = \begin{bmatrix} \mathbf{CB} & 0 & 0 & \dots & 0 \\ \mathbf{CAB} & \mathbf{CB} & 0 & \dots & 0 \\ \mathbf{CA}^2\mathbf{B} & \mathbf{CAB} & \mathbf{CB} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{CA}^{N_p-1}\mathbf{B} & \mathbf{CA}^{N_p-2}\mathbf{B} & \mathbf{CA}^{N_p-3}\mathbf{B} & \dots & \mathbf{CA}^{N_p-N_c}\mathbf{B} \end{bmatrix} \end{aligned} \quad (3.62)$$

The aim of the model predictive control system is to minimize the disparity between the predicted output and the reference signal. We operate under the assumption that the set-point signal remains steady within the optimization window. This goal is then translated into a strategy to determine the optimal control parameter vector,  $\Delta\mathbf{U}$ . Considering  $r(k)$  as reference signal and defining  $\mathbf{R}_s^T$  as the vector:

$$\mathbf{R}_s^T = \mathbf{1}^{1 \times N_p} r(k_i) \quad (3.63)$$

we can define the cost function:

$$J = (\mathbf{R}_s^T - \mathbf{Y})^T (\mathbf{R}_s^T - \mathbf{Y}) + \Delta\mathbf{U}^T \bar{\mathbf{R}} \Delta\mathbf{U}. \quad (3.64)$$

$\bar{\mathbf{R}}$  is a diagonal matrix of size  $N_c \times N_c$  and has the role of design parameters that has to be tuned considering a trade off between performances and size of  $\Delta\mathbf{U}$ .

Now that we have introduced the objective function of the optimization problem we can define constraints. Different kinds of constraints can be introduced:

- Constraints on the control variable incremental variation  $\Delta u(k)$
- Constraints on the amplitude of the control variable  $u(k)$
- Constraints on output  $y(k)$

For our purposes, requirements impose constraints only on control variable  $u(k)$ :

$$u^{min} \leq u(k_i) \leq u^{max}. \quad (3.65)$$

Considering the fact that:

$$\Delta u(k_i) = u(k_i) - u(k_i - 1) \quad (3.66)$$

we can express constraints on  $u(k_i)$  in terms of  $\Delta u(k)$  as:

$$\begin{bmatrix} 1 & \mathbf{0}^{1 \times N_c - 1} \\ -1 & \mathbf{0}^{1 \times N_c - 1} \end{bmatrix} \Delta \mathbf{U}(k_i) \leq \begin{bmatrix} u^{max} - u(k_i - 1) \\ -u^{min} + u(k_i - 1) \end{bmatrix}. \quad (3.67)$$

At the end we obtained the following optimization problem:

$$\begin{aligned} \min_{\Delta \mathbf{U}} & (\mathbf{R}_s^T - \mathbf{Y})^T (\mathbf{R}_s^T - \mathbf{Y}) + \Delta \mathbf{U}^T \bar{\mathbf{R}} \Delta \mathbf{U} \\ \text{s.t.} & \begin{bmatrix} 1 & \mathbf{0}^{1 \times (N_c - 1)} \\ -1 & \mathbf{0}^{1 \times (N_c - 1)} \end{bmatrix} \Delta \mathbf{U}(k_i) \leq \begin{bmatrix} u^{max} - u(k_i - 1) \\ -u^{min} + u(k_i - 1) \end{bmatrix}. \end{aligned} \quad (3.68)$$

It is a constrained Quadratic Programming (QP), hence an optimization problem where we minimize a quadratic objective in a polyhedron. It and can be written in standard form, considering equation 3.61:

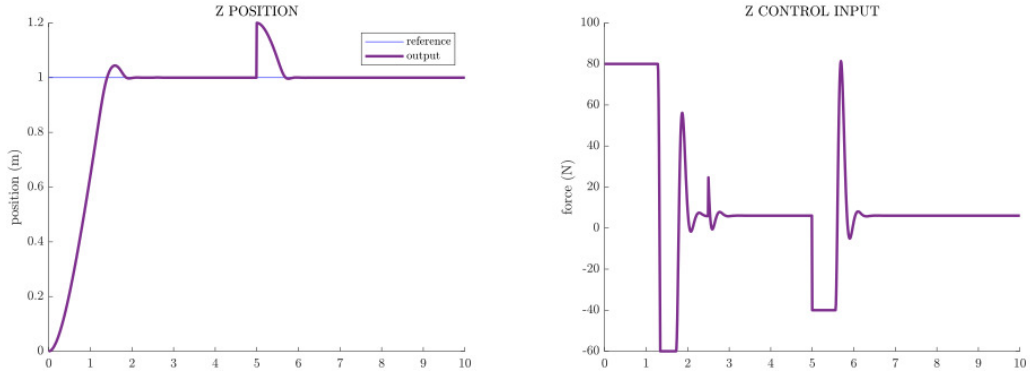
$$\begin{aligned} \min_{\Delta \mathbf{U}} & \Delta \mathbf{U}^T 2(\Phi^T \Phi \bar{\mathbf{R}}) \Delta \mathbf{U} - 2\Phi^T (\mathbf{R}_s - \mathbf{F}\mathbf{x}(k_i)) \Delta \mathbf{U} \\ \text{s.t.} & \begin{bmatrix} 1 & \mathbf{0}^{1 \times (N_c - 1)} \\ -1 & \mathbf{0}^{1 \times (N_c - 1)} \end{bmatrix} \Delta \mathbf{U}(k_i) \leq \begin{bmatrix} u^{max} - u(k_i - 1) \\ -u^{min} + u(k_i - 1) \end{bmatrix}. \end{aligned} \quad (3.69)$$

In general if the Hessian of the objective is positive semi-definite, then the problem is tractable and efficiently solvable since it is convex. The design of MPC for heave, roll and pitch dynamic of EVA is based on the procedure explained before; the design parameter  $\bar{\mathbf{R}}$  was tuned in each controller to fit requirements. At each time instant, we computed the objective function and verified that its Hessian is positive semi-definite to ensure the tractability of the problem. Finally, employing the receding horizon principle, we applied the control law to the plant. The algorithm used for solving the quadratic program is the active-set algorithm, that is in general one of the best methods for small/medium scale problems. The algorithm starts by trying to guess the optimal active set and then, if the guess is not correct, repeatedly drop one index from the actual estimation of the set using information from gradient and Lagrange multipliers [30]. At the end the control law  $u(t)$  is the solution of the following problem:

$$\begin{aligned} u(k) &= u(k - 1) + \arg \min_{\Delta \mathbf{U}} \Delta \mathbf{U}^T 2(\Phi^T \Phi \bar{\mathbf{R}}) \Delta \mathbf{U} - 2\Phi^T (\mathbf{R}_s - \mathbf{F}\mathbf{x}(k_i)) \Delta \mathbf{U} \\ \text{s.t.} & \begin{bmatrix} 1 & \mathbf{0}^{1 \times (N_c - 1)} \\ -1 & \mathbf{0}^{1 \times (N_c - 1)} \end{bmatrix} \Delta \mathbf{U}(k_i) \leq \begin{bmatrix} u^{max} - u(k_i - 1) \\ -u^{min} + u(k_i - 1) \end{bmatrix}. \end{aligned} \quad (3.70)$$

The result of simulations on control oriented models are in the next figures. For what regard heave, the reference signal is a step signal of amplitude 1  $m$ . At time

$t = 2.5$  s we introduced a step disturbances  $d_1 = 20$  N, while at time  $t = 5$  s a step disturbances  $d_2 = 0.2$  m. The output response and the control input are plotted in figure 3.16.

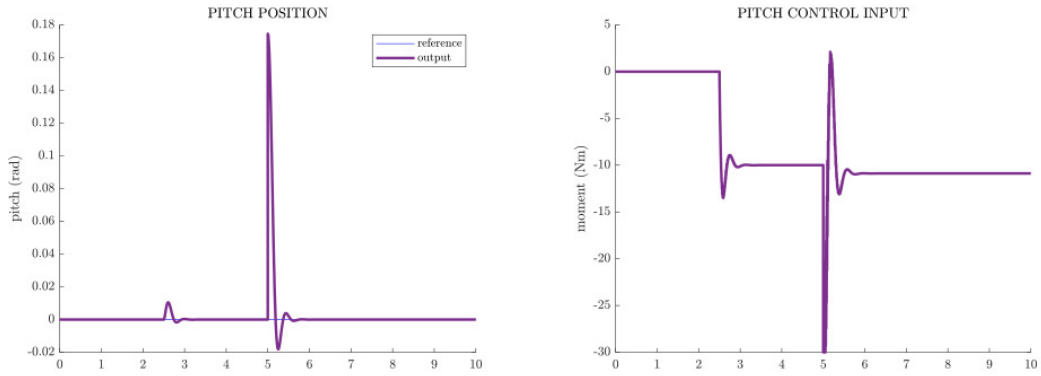


**Figure 3.16:** MPC for heave dynamic

As we can see the requirements are satisfied with a safety margin since:

$$\hat{s} = 0.044 \text{ m}; \quad t_{s,1\%} = 1.82 \text{ s}; \quad |y_{d_1, d_2}^\infty| = 0. \quad (3.71)$$

For pitch control, in a first simulation, the reference signal is a constant signal of amplitude zero. At time  $t = 2.5$  s we introduced a step disturbances  $d_1 = 10$  Nm, while at time  $t = 5$  s a step disturbances  $d_2 = 10$  degree. The output response and the control input are plotted in figure 3.17.

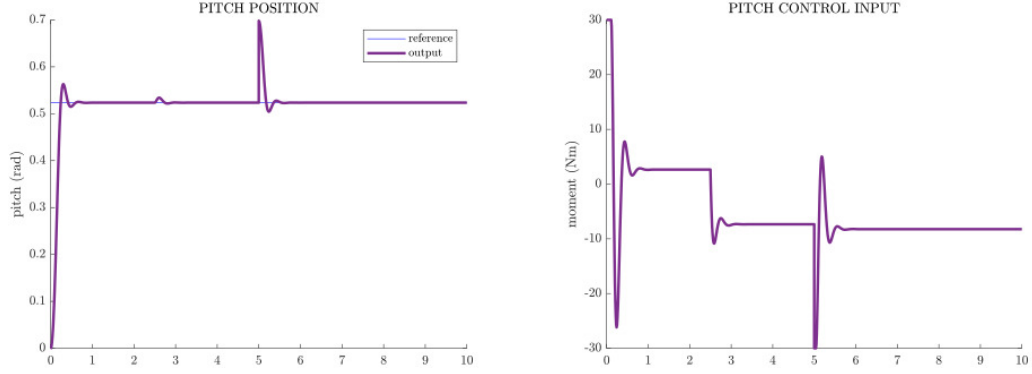


**Figure 3.17:** MPC regulation for pitch dynamic

As we can see the requirements are satisfied with a safety margin since:

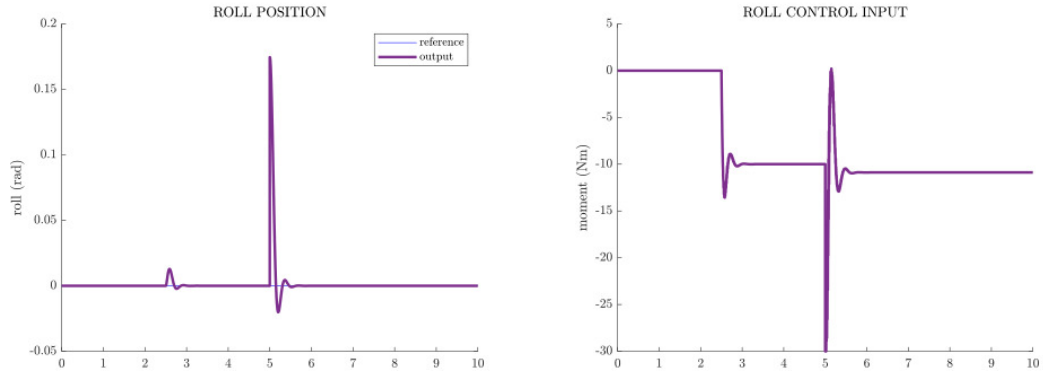
$$|y_{d_1, d_2}^\infty| = 0; \quad \max y_{d_1} = 0.010 \text{ rad}; \quad t_{s,1\%}^{d_1} = 0.12 \text{ s}; \quad t_{s,1\%}^{d_2} = 0.31 \text{ s}. \quad (3.72)$$

In a second simulation everything is left unchanged except for reference signal that is now an angle of  $\frac{\pi}{6}$  rad. As we can see in the next figures, tracking is achieved.



**Figure 3.18:** MPC tracking for pitch dynamic

For roll control, the reference signal is a constant signal of amplitude zero. At time  $t = 2.5$  s we introduced a step disturbances  $d_1 = 10$  Nm, while at time  $t = 5$  s a step disturbances  $d_2 = 10$  degree. The output response and the control input are plotted in figure 3.19.



**Figure 3.19:** MPC for roll dynamic

As we can see the requirements are satisfied with a safety margin since:

$$|y_{d_1, d_2}^\infty| = 0; \quad \max y_{d_1} = 0.013 \text{ rad}; \quad t_{s,1\%}^{d_1} = 0.13 \text{ s}; \quad t_{s,1\%}^{d_2} = 0.26 \text{ s}. \quad (3.73)$$

We can notice that, unlike what occurred in other types of control, input saturation requirements are satisfied in all the control laws simulated above. That's because we considered constraints in the optimization problem involved at each time instant as in equation 3.70.

## Chapter 4

# Control Performance Comparison

In this chapter, the performances of the various control systems developed in chapter 3 will be compared by simulating the behavior of EVA using the simulation model (Fossen model) presented in section 2.1. The control achieved through pole placement will be presented in its original form and in anti-windup form (subsection 3.1.1), the LQR will be simulated in its original form, while the MPC, considering that it takes into account the constraints of the control input within its design equations, will be simulated in its original form.

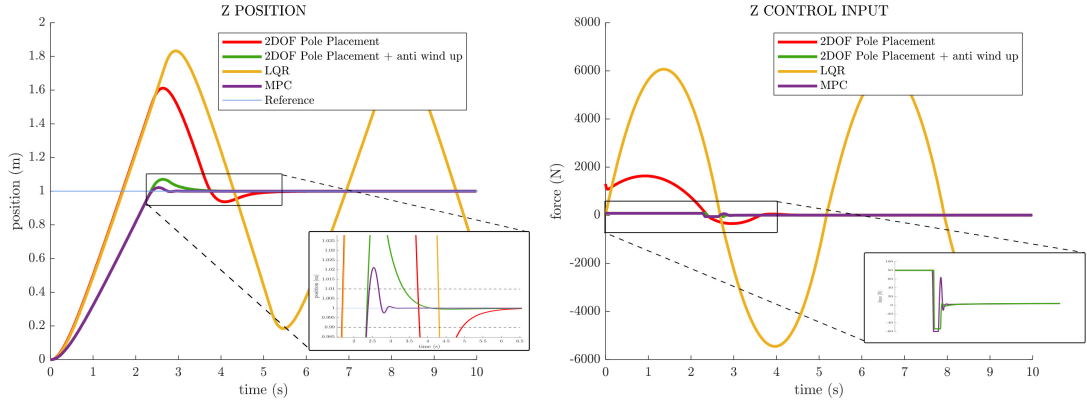
In the first section we will simply report all the performances on the simulator, while in section 4.2 a critical analysis of the results is presented: in that section, we will try to understand which of the implemented controllers is the best and the compromises that need to be addressed if we are looking for simplicity or if we are looking for performances and robustness.

### 4.1 Performance comparison on simulator

In this subsection we will report all the control laws performances. The problems analyzed are the ones reported in figure 3.2, where from physical observations we understood the various control problems.

All the results are simulated in MATLAB/Simulink and the mathematical model used in the closed loop scheme is the Fossen model presented in section 2.1.

For what regard heave tracking, we ran a 10 *s* simulation with a 1 *m* step reference signal. The output response and the control input for each one of the control laws are plotted in figure 4.1.



**Figure 4.1:** Heave tracking performance comparison

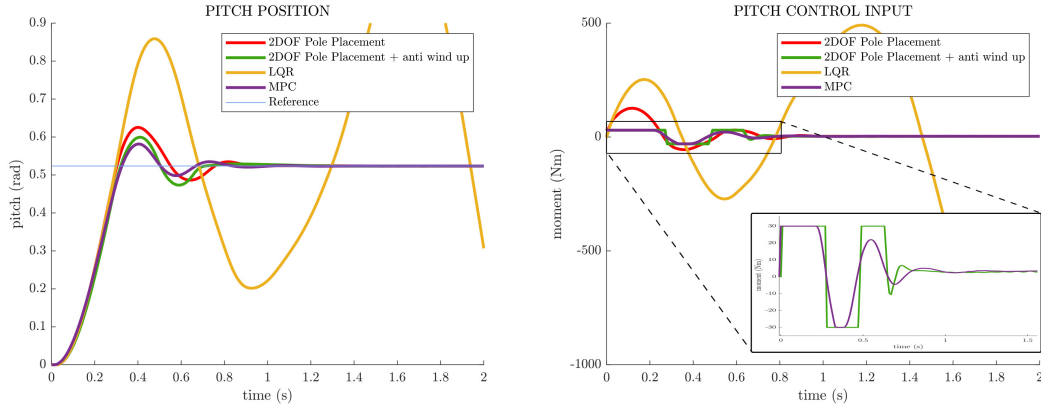
As we can easily see from the plots, the LQR controller has serious windup problems and input constraint violation, so it is not able to achieve the requirements. The 2 DOF Pole placement control in its original form is able to track the reference, but it violates constraint. The problem is solved introducing the anti-windup scheme as we can see from the green line. Hence the 2 DOF Pole Placement control with anti-windup scheme is able to ensure heave tracking with a safety margin on requirements since:

$$\hat{s} = 0.070 \text{ m}; \quad t_{s,1\%} = 3.35 \text{ s}; \quad -60 \text{ N} \leq u(t) \leq 80 \text{ N}. \quad (4.1)$$

The MPC is the one with the best performance and complies with the input constraints:

$$\hat{s} = 0.021 \text{ m}; \quad t_{s,1\%} = 2.67 \text{ s}; \quad -60 \text{ N} \leq u(t) \leq 80 \text{ N}. \quad (4.2)$$

For what regard pitch tracking, we ran a 10 s simulation with a  $\frac{\pi}{6}$  rad step reference signal. The output response and the control input for each one of the control laws are plotted in figure 4.2.



**Figure 4.2:** Pitch tracking performance comparison

As we can easily see from the plots, also in this case the LQR controller has serious windup problems and input constraint violation, so it is not able to achieve the requirements. As well as the heave tracking problem, the 2 DOF Pole placement control in its original form is able to track the reference, but it violates constraint. The problem is solved introducing the anti-windup scheme as we can see from the green line. Hence the 2 DOF Pole Placement control with anti-windup scheme is able to ensure heave tracking with a safety margin on requirements since:

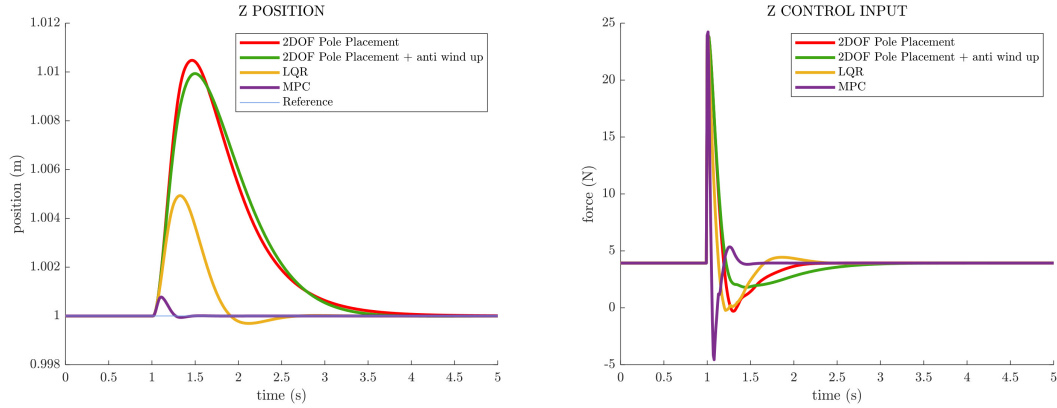
$$\hat{s} = 0.0764 \text{ rad}; \quad t_{s,1\%} = 0.90 \text{ s}; \quad -30 \text{ Nm} \leq u(t) \leq 30 \text{ Nm}. \quad (4.3)$$

The MPC is the one with the best performance and complies with the input constraints:

$$\hat{s} = 0.0564 \text{ rad}; \quad t_{s,1\%} = 0.79 \text{ s}; \quad -30 \text{ Nm} \leq u(t) \leq 30 \text{ Nm}. \quad (4.4)$$

Lets now consider disturbance rejection problems starting from heave dynamic. For heave disturbance rejection controller comparison, we runned a 5 s simulation starting from a condition in wich the ROV has already achieved a 1 m step reference signal. At time  $t = 1 \text{ s}$  a step disturbance of 20 N is introduced. The output response and the control input for each one of the control laws are plotted in figure 4.3.

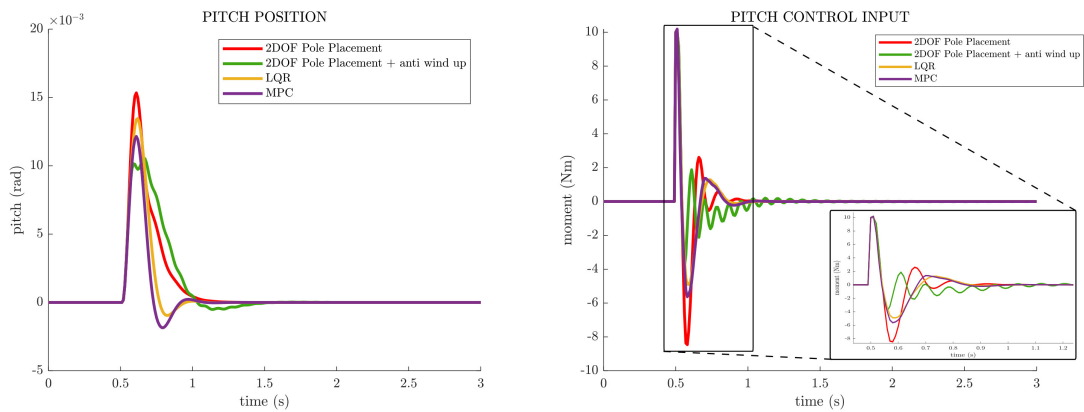




**Figure 4.3:** Heave disturbance rejection performance comparison

We can notice that all the control laws are able to counteract disturbance and to satisfy input saturation constraint. So, this time the differences between all the controllers are simply performances. Given that all controllers are good enough to allow a deviation of no more than one percent from the reference, then the only meaningful parameter to use for comparison is the maximum output value reached. The worst is the 2 DOF Pole Placement, that, by switching to the anti-windup form, reduces the maximum from 0.011  $m$  to 0.0099  $m$ . The maximum in the LQR case is 0.0055  $m$ , but the best performance is achieved with the MPC control law, that reach a maximum of 0.0008  $m$

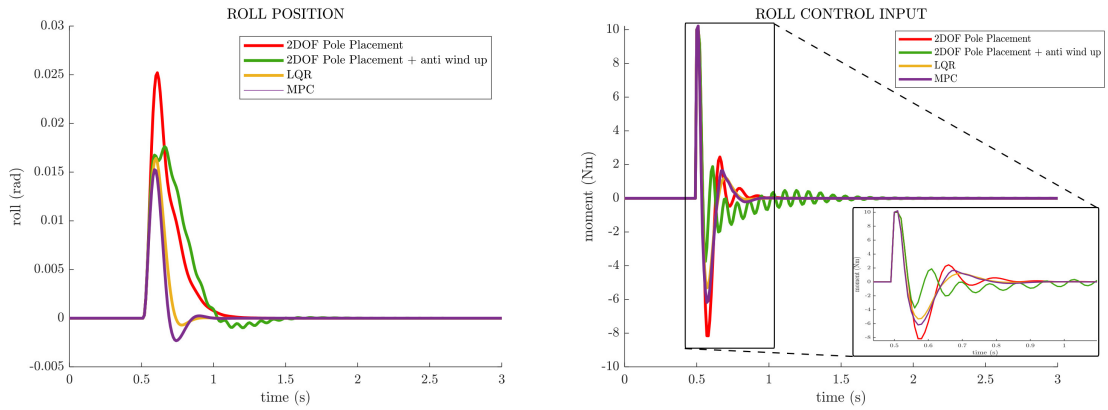
For pitch disturbance rejection controller comparison, we runned a 3  $s$  simulation where at time  $t = 0.5 s$  a step disturbance of 10  $Nm$  is introduced. The output response and the control input for each one of the control laws are plotted in figure 4.4.



**Figure 4.4:** Pitch disturbance rejection performance comparison

We can notice that all the control laws are able to counteract disturbance and to satisfy input saturation constraint. So, as for heave disturbance compensation, also this time the differences between all the controllers are simply performances. The worst control law is the 2 DOF Pole Placement, that has maximum pitch displacement from reference of  $0.015 \text{ rad}$ . The maximum in the LQR case is  $0.014 \text{ rad}$ , while for the MPC control law it is  $0.012 \text{ rad}$ . This time the best performance is achieved from the 2 DOF Pole Placement control law in anti-windup form with a maximum displacement of  $0.011 \text{ rad}$ .

For roll disturbance rejection controller comparison, we runned a  $3 \text{ s}$  simulation where at time  $t = 0.5 \text{ s}$  a step disturbance of  $10 \text{ Nm}$  is introduced. The output response and the control input for each one of the control laws are plotted in figure 4.5.



**Figure 4.5:** Roll disturbance rejection performance comparison

We can notice that all the control laws are able to counteract disturbance and to satisfy input saturation constraint. So, as for heave disturbance compensation, also this time the differences between all the controllers are simply performances. The worst is the 2 DOF Pole Placement, that, by switching to the anti-windup form, reduces the maximum from  $0.025 \text{ rad}$  to  $0.018 \text{ rad}$ . The maximum in the LQR case is  $0.017 \text{ m}$ , but the best performance is achieved with the MPC control law, that reach a maximum roll displacement of  $0.015 \text{ rad}$ .


## 4.2 Critical analysis

After understanding the reasons behind the necessity of a control system in an ROV, and after detailing how to develop mathematical models for simulation purposes and control-oriented ones, our focus shifted to the actual design of various control systems, transitioning from more traditional methods such as pole placement to optimal control methods like MPC. Finally, in the just-concluded section, we observed the results of each of these controllers on the EVA ROV simulator, showcasing the strengths and weaknesses of each. At this point, it's time for a critical analysis of the results obtained to understand which control law is most suitable.

For a correct choice, it's necessary to ask the right questions and understand various factors of the system to be controlled. The three fundamental aspects are: what level of performance do we aim to achieve, what computational complexity can we handle online in real time, and how faithful is our simulator, and therefore how much can we rely on the data obtained by simulating the dynamic behavior of our ROV. We will explain how, on the basis of this aspects, we can choose a control law for a generic ROV or for our specific ROV EVA.

All the characteristic of our control systems are summarized in the following table reported in figure 4.6.

### CRITICAL ANALISYS

	POLE PLACEMENT	POLE PLACEMENT + ANTI WINDUP	LQR	MPC
DESIGN EFFORT				
PERFORMANCES ON REQUIREMENTS	<input checked="" type="checkbox"/> Tracking <input checked="" type="checkbox"/> Disturbance rejection <input checked="" type="checkbox"/> Constraints	<input checked="" type="checkbox"/> Tracking <input checked="" type="checkbox"/> Disturbance rejection <input checked="" type="checkbox"/> Constraints	<input checked="" type="checkbox"/> Tracking <input checked="" type="checkbox"/> Disturbance rejection <input checked="" type="checkbox"/> Constraints	<input checked="" type="checkbox"/> Tracking <input checked="" type="checkbox"/> Disturbance rejection <input checked="" type="checkbox"/> Constraints
COMPUTATIONAL COMPLEXITY				

**Figure 4.6:** Critical comparison of controllers

In this table we reported the three aspect listed before, plus and additional one that put emphasis on the design effort needed.

At first glance at the table, it is immediately evident that LQR should be

excluded from the control selection. In fact, it fails to meet the requirements due to a significant windup effect when it's asked to track a specific signal. The 2 DOF controller obtained through Pole Placement shows slightly better performance and is able to track signals, but it also exhibits a slight windup due to not adhering to the constraints of the control input. This problem is overcome in the 2 DOF controller obtained through Pole Placement with an anti-windup scheme, as well as in MPC, which directly considers constraints within its optimization problem. Among all, the controller with the best performance is indeed MPC, thanks to its optimal control nature.

Speaking instead about the computational complexity required by the ROV in real time, MPC ranks at the bottom of the list. It requires solving a significant optimization problem in real time and online for each time step. Obviously, this demands high computational power from the ROV, and it's not guaranteed that it can provide it. On the other hand, the other controllers are all fairly simple from a computational standpoint: the two controllers obtained with Pole Placement are simple transfer functions, while LQR consists of coefficients calculated by solving an offline optimization problem.

Finally, speaking of the design aspect, both LQR and MPC require greater design effort compared to 2 DOF Pole Placement, suggesting the use of the latter for rapid prototyping.

Based on what has been said so far, the choice lies between the 2-DOF controller with anti-windup scheme and MPC because others have problems of performance. If our priority is to have a controller with acceptable performance, lightweight computationally, and simple to implement, then the controller obtained through Pole Placement with anti-windup scheme is undoubtedly the best choice. However, if the priority is to have a controller with very high performance, especially if we believe that our simulation model is not sufficiently faithful to reality, then the choice falls on MPC, assuming that the ROV is capable of handling the computational complexity required by such a controller.

In the case of the EVA ROV, numerous efforts have been made to create the simulation model. In fact, the CAD used to derive the mechanical parameters is highly faithful to reality, as are the hydrodynamic parameters obtained through CFD, which were acquired after extensive hours of study and simulations. This is why we can assert that our simulator is truly a high-fidelity model of the EVA. Consequently, we have opted for controller, simpler both in terms of design and computational demands on the ROV, but at the same time ensuring good tracking and disturbance rejection performance. The controller we're talking about is, of course, the 2 DOF control obtained through Pole Placement with an anti-windup scheme.

# Conclusions

The aim of this thesis is the mathematical modeling and control of a ROV. The purpose of the control is to stabilize and enhance the maneuverability of the ROV at different depths. This is because the ROV must be capable of performing various manipulations with a robotic arm, and it is therefore crucial for it to automatically maintain the desired depth and to compensate for disturbances caused by interaction with external objects. So, the objective of the control was to track a desired depth or a specific pitch angle to facilitate operations on the seabed, and to compensate for disturbances on the heave, pitch, and roll dynamics.

After discussing the main features and utilities of ROVs and presenting the ROV used for this thesis, the focus of the work shifted to mathematical modeling. In fact, in Chapter 2, two types of mathematical models were presented: the first is a high-fidelity model useful for accurately simulating the dynamic behavior of the ROV (section 2.1), while the others are control-oriented models (section 2.3), thus, linearized models that represent individual dynamics in a simplified manner. These latter models are derived from the high-fidelity model based on some assumptions and simplifications grounded in the physics of the system.

In Chapter 3, the design of various control systems based on control-oriented models was discussed. First, a 2 DOF controller was developed using Pole Placement, explaining also how to handle the physical limits of thrust saturation and thus avoid the destructive problem of windup. Subsequently, the attention shifted to optimal control, and LQR and MPC controllers were implemented.

Finally, in Chapter 4, the performance of each controller in closed loop with the high-fidelity model was compared, and a critical analysis was presented highlighting the advantages of one type of control over another.

Ultimately, it emerged that an MPC controller ensures the best performance, although it requires lot of design effort and computational complexity. It requires solving an online optimization problem in real-time for each time instant. Hence, the suggestion of this thesis is to use a 2 DOF controller obtained with Pole Placement and an anti-windup scheme. Indeed, it represents the right compromise between performance, design effort and online computational cost. It is simple to design and lightweight computationally. Moreover, it demonstrates good performance in

the simulator.

Indeed, one of the possible advancements that could be made is the transition to data-driven black box models. They could be more faithful and representative of the real dynamics of the system, thus providing a solid basis for the design of controllers like those developed in this thesis. This is indeed one of the ongoing projects that I am carrying out together with the Control System Division within the PoliTOcean team. Our goal is to achieve a robust identification of different control-oriented models from measurements obtained from our IMU and our barometer. This work, along with the implementation of additional control systems, could lead to further improvement in performance and consequently to a better performance of our EVA ROV in future competitions.

# Bibliography

- [1] Romano Capocci, Gerard Dooly, Edin Omerdić, Joseph Coleman, Thomas Newe, and Daniel Toal. «Inspection-Class Remotely Operated Vehicles—A Review». In: *Journal of Marine Science and Engineering* 5.1 (Mar. 2017), p. 13. ISSN: 2077-1312. DOI: 10.3390/jmse5010013. URL: <http://dx.doi.org/10.3390/jmse5010013> (cit. on p. 1).
- [2] Amit Shukla and Hamad Karki. «Application of robotics in offshore oil and gas industry— A review Part II». In: *Robotics and Autonomous Systems* 75 (Jan. 2016), pp. 508–524. ISSN: 0921-8890. DOI: 10.1016/j.robot.2015.09.013. URL: <http://dx.doi.org/10.1016/j.robot.2015.09.013> (cit. on p. 2).
- [3] R.D. Christ and Robert Wernli. «The ROV Manual: A User Guide for Remotely Operated Vehicles: Second Edition». In: *The ROV Manual: A User Guide for Remotely Operated Vehicles: Second Edition* (Jan. 2013), pp. 1–679 (cit. on p. 2).
- [4] Thor Fossen. *Handbook of Marine Craft Hydrodynamics and Motion Control*. Apr. 2021. ISBN: 978-1119575047. DOI: 10.1002/9781119994138 (cit. on pp. 3, 5, 8, 9, 16, 17, 19, 20, 29).
- [5] S.M. Ahmad and R. Sutton. «Dynamic Modelling of a Remotely Operated Vehicle». In: *IFAC Proceedings Volumes* 36.4 (Apr. 2003), pp. 43–48. ISSN: 1474-6670. DOI: 10.1016/S1474-6670(17)36655-7. URL: [http://dx.doi.org/10.1016/S1474-6670\(17\)36655-7](http://dx.doi.org/10.1016/S1474-6670(17)36655-7) (cit. on p. 3).
- [6] C. S. Chin, W. P. Lin, and J. Y. Lin. «Experimental validation of open-frame ROV model for virtual reality simulation and control». In: *Journal of Marine Science and Technology* 23.2 (July 2017), pp. 267–287. ISSN: 1437-8213. DOI: 10.1007/s00773-017-0469-3. URL: <http://dx.doi.org/10.1007/s00773-017-0469-3> (cit. on pp. 3, 26, 29).
- [7] Ahsan Tanveer and S.M Ahmad. «Heave Modeling and Control of a Micro-ROV». In: *2021 International Conference on Robotics and Automation in Industry (ICRAI)*. IEEE, Oct. 2021. DOI: 10.1109/icrai54018.2021.9651465. URL: <http://dx.doi.org/10.1109/ICRAI54018.2021.9651465> (cit. on p. 3).

- [8] Ahsan Tanveer and Sarvat M. Ahmad. «High fidelity modelling and GA optimized control of yaw dynamics of a custom built remotely operated unmanned underwater vehicle». In: *Ocean Engineering* 266 (Dec. 2022), p. 112836. ISSN: 0029-8018. DOI: 10.1016/j.oceaneng.2022.112836. URL: <http://dx.doi.org/10.1016/j.oceaneng.2022.112836> (cit. on pp. 3, 29).
- [9] Junjie Dong and Xingguang Duan. «A Robust Control via a Fuzzy System with PID for the ROV». In: *Sensors* 23.2 (Jan. 2023), p. 821. ISSN: 1424-8220. DOI: 10.3390/s23020821. URL: <http://dx.doi.org/10.3390/s23020821> (cit. on p. 3).
- [10] Enrico Anderlini, Gordon G. Parker, and Giles Thomas. «Control of a ROV carrying an object». In: *Ocean Engineering* 165 (Oct. 2018), pp. 307–318. ISSN: 0029-8018. DOI: 10.1016/j.oceaneng.2018.07.022. URL: <http://dx.doi.org/10.1016/j.oceaneng.2018.07.022> (cit. on pp. 3, 29).
- [11] Malte von Benzon, Fredrik Fogh Sørensen, Esben Uth, Jerome Jouffroy, Jesper Liniger, and Simon Pedersen. «An Open-Source Benchmark Simulator: Control of a BlueROV2 Underwater Robot». In: *Journal of Marine Science and Engineering* 10.12 (Dec. 2022), p. 1898. ISSN: 2077-1312. DOI: 10.3390/jmse10121898. URL: <http://dx.doi.org/10.3390/jmse10121898> (cit. on p. 3).
- [12] Zongsheng Wang, Yue Liu, Zhendong Guan, and Yan Zhang. «An Adaptive Sliding Mode Motion Control Method of Remote Operated Vehicle». In: *IEEE Access* 9 (2021), pp. 22447–22454. ISSN: 2169-3536. DOI: 10.1109/access.2021.3055204. URL: <http://dx.doi.org/10.1109/ACCESS.2021.3055204> (cit. on p. 3).
- [13] Aldo-Jonathan Muñoz-Vázquez, Heriberto Ramírez-Rodríguez, Vicente Parra-Vega, and Anand Sánchez-Orta. «Fractional sliding mode control of underwater ROVs subject to non-differentiable disturbances». In: *International Journal of Control, Automation and Systems* 15.3 (Mar. 2017), pp. 1314–1321. ISSN: 2005-4092. DOI: 10.1007/s12555-015-0210-0. URL: <http://dx.doi.org/10.1007/s12555-015-0210-0> (cit. on p. 3).
- [14] Mai The Vu, Tat-Hien Le, Ha Le Nhu Ngoc Thanh, Tuan-Tu Huynh, Mien Van, Quoc-Dong Hoang, and Ton Duc Do. «Robust Position Control of an Over-actuated Underwater Vehicle under Model Uncertainties and Ocean Current Effects Using Dynamic Sliding Mode Surface and Optimal Allocation Control». In: *Sensors* 21.3 (Jan. 2021), p. 747. ISSN: 1424-8220. DOI: 10.3390/s21030747. URL: <http://dx.doi.org/10.3390/s21030747> (cit. on p. 3).
- [15] URL: <https://bluerobotics.com/store/thrusters/t100-t200-thrusters/t200-thruster-r2-rp/> (cit. on pp. 3, 25).
- [16] URL: <https://witmotion-sensor.com/> (cit. on p. 3).



- [17] URL: <https://bluerobotics.com/store/sensors-cameras/sensors/bar02-sensor-r1-rp/> (cit. on p. 3).
- [18] Gianluca Antonelli. *Underwater Robots*. Springer International Publishing, 2014. ISBN: 9783319028774. DOI: 10.1007/978-3-319-02877-4. URL: <http://dx.doi.org/10.1007/978-3-319-02877-4> (cit. on pp. 5, 9, 16–18, 20).
- [19] Aleksey Kabanov, Vadim Kramar, and Igor Ermakov. «Design and Modeling of an Experimental ROV with Six Degrees of Freedom». In: *Drones* 5.4 (2021). ISSN: 2504-446X. DOI: 10.3390/drones5040113. URL: <https://www.mdpi.com/2504-446X/5/4/113> (cit. on p. 23).
- [20] Chu-Jou Wu. «6-DoF Modelling and Control of a Remotely Operated Vehicle». M. Eng. thesis. Flinders University, 2018 (cit. on pp. 23, 26).
- [21] Persing Cardenas and Ettore A. de Barros. «Estimation of AUV Hydrodynamic Coefficients Using Analytical and System Identification Approaches». In: *IEEE Journal of Oceanic Engineering* 45.4 (Oct. 2020), pp. 1157–1176. ISSN: 2373-7786. DOI: 10.1109/joe.2019.2930421. URL: <http://dx.doi.org/10.1109/JOE.2019.2930421> (cit. on p. 26).
- [22] Graham C. Goodwin, Stefan F. Graebe, and Mario E. Salgado. *Control System Design*. 1st. USA: Prentice Hall PTR, 2000. ISBN: 0139586539 (cit. on pp. 39, 40, 46).
- [23] Michel Kinnaert and Vincent Blondel. «Discrete-time pole placement with stable controller». In: *Automatica* 28.5 (Sept. 1992), pp. 935–943. ISSN: 0005-1098. DOI: 10.1016/0005-1098(92)90146-7. URL: [http://dx.doi.org/10.1016/0005-1098\(92\)90146-7](http://dx.doi.org/10.1016/0005-1098(92)90146-7) (cit. on p. 39).
- [24] R. Vilanova. «Reference controller design in 2-DOF control: Quadratic cost optimisation». In: *Electrical Engineering* 90.4 (June 2007), pp. 275–281. ISSN: 1432-0487. DOI: 10.1007/s00202-007-0075-1. URL: <http://dx.doi.org/10.1007/s00202-007-0075-1> (cit. on p. 39).
- [25] Samet Ahmed and Kourid Yahia. «Implementation of Integral LQR Controller with Anti-Windup on FPGA». In: *2019 1st International Conference on Sustainable Renewable Energy Systems and Applications (ICSRESA)*. IEEE, Dec. 2019. DOI: 10.1109/icsresa49121.2019.9182704. URL: <http://dx.doi.org/10.1109/ICSRESA49121.2019.9182704> (cit. on p. 46).
- [26] Saeed Al-Haddad and Herman Wahid. «DECOUPLED INTEGRAL LQR CONTROLLER WITH ANTI- WINDUP COMPENSATOR FOR MIMO TWO ROTOR AERODYNAMICAL SYSTEM (TRAS)». In: *Journal of Engineering Science and Technology* 14 (June 2019), pp. 1374–1397 (cit. on p. 46).

- [27] Andrew Phillips and Ferat Sahin. «Optimal control of a twin rotor MIMO system using LQR with integral action». In: Aug. 2014, pp. 114–119. DOI: 10.1109/WAC.2014.6935709 (cit. on p. 46).
- [28] David Di Ruscio. «Model Predictive Control with Integral Action: A simple MPC algorithm». In: *Modeling, Identification and Control: A Norwegian Research Bulletin* 34.3 (2013), pp. 119–129. ISSN: 1890-1328. DOI: 10.4173/mic.2013.3.2. URL: <http://dx.doi.org/10.4173/mic.2013.3.2> (cit. on p. 50).
- [29] Liuping Wang. *Model predictive control system design and implementation using MATLAB (R)*. en. 2009th ed. Advances in industrial control. London, England: Springer, Mar. 2009 (cit. on p. 50).
- [30] Jorge Nocedal and Stephen J Wright. *Numerical Optimization*. en. Ed. by Jorge Nocedal and Stephen J Wright. 1st ed. Springer Series in Operations Research and Financial Engineering. New York, NY: Springer, June 2000 (cit. on p. 54).