POLITECNICO DI TORINO

Master's Degree in Mechatronic Engineering

Master's Degree Thesis

IDENTIFICATION OF ZENO AUV MODEL PARAMETERS

ASSOCIATED COMPANY: **MDM TEAM srl**

SUPERVISOR:
Prof. **Marcello Chiaberge**

COMPANY SUPERVISORS:
Ing. **Vincenzo Calabrò**
Ing. **Niccolò Monni**

CANDIDATE:
Lorenzo Sacco
Matr:251480

Accademic year 2019/2020

Contents

List of Figures

Abstract

My master thesis project has been developed during my internship in "MDM Team", an Italian company specialized in underwater robotics.

My task was to identify the model parameters for an Autonomous Underwater Vehicle (AUV) called Zeno. Indeed, the dynamic behaviour of the vehicle can be described through a suitable mathematical model whose parameters are generally unknown. For this reason, it is necessary to find a good estimation of their values. In order to get this result I formulated a sequence of optimization problems. Each of these is formulated in order to return part of the complete set of parameters to be identified. The estimated values obtained by solving each problem are used then as inputs for the following problems, together with the sensors data collected during specific tests performed in a testing pool.
The identified model has been then used in order to add a model-based feedforward action, called Computed Torque, to the already implemented PID motion controller. This is a key point for the development of the control system. The PID controller alone ensures stability, while a controller action based on the dynamic model of the vehicle is needed to ensure better performances in the trajectory execution.

Mentions about the AUVs state of the art and about theoretical notions are presented in Chapter 1, together with a more detailed explanation of the motivations that led to my work of thesis. In Chapter 2 the model used to describe the dynamics of Zeno is illustrated. Chapter 3 is the core of this dissertation, where the identification algorithm is shown and explained. Chapter 4 presents the results obtained applying this estimation process, with all the necessary comments. The Computed Torque action based on the outcome of my work has been already implemented by the "MDM Team". Chapter 5 illustrates the structure of the controller and the improvements achieved in the navigation control. In Appendix A few details about the code structure and its Graphic User Interface is shown.

Chapter 1

Introduction

1.1 AUVs State of the Art

An Autonomous Underwater Vehicle (AUV) is an unmanned system meant to solve underwater tasks without the intervention of a human operator. Starting from the 1960s, universities, institutes and governments began to theorize and experiment AUV technologies. However, only in the last two decades the interest on this field brought to a stable development at a quite fast rate [1].
AUVs are now used in several applications, such us archaeological research, mapping of the sea bottom, harbour security, searching for sea-mines, mines detecting and many others. The advantage of using this kind of vehicle is achieving tasks too dangerous or impossible for humans, sometimes at a lower cost than using a human operated vehicle.
Anyway, underwater robotics is a really challenging branch; indeed lot of problems still have to be solved or improved. For example, autonomy and charge/change of the battery is a critical point, due to the hard environment as well as communication due to the strong attenuation of radio waves inside the water. Moreover, environment disturbances like currents or waves can be very significant and hardly affecting the dynamic behavior of a vehicle.

In Europe several companies are researching and producing AUV vehicles. In Norway, Kongsberg Maritime Ltd. is producing in large scale a AUV named 'HUGIN' [2] (Figure 1.1a). This torpedo shaped vehicle is available in different versions, according to its depth range; the sensing concept allows alternate sensors for geological survey, search operations etc. Another example is the 'GAVIA' (Figure 1.1b) from Hafmynd company in Iceland [3]. This cylindrical AUV is available for commercial, scientific and defense application [4].

(a) Hugin AUV

(b) Gavia Offshore Surveyor

Figure 1.1: Two European AUV Example

As stated before, I did my internship in "MDM Team", an official spin-off company of the University of Florence, founded in 2012 by a group of professors and researchers of the Department of Industrial Engineering. The aim of the Team is to develop high-tech innovative robotics products and they are specialized in the field of underwater robotics [5]. I have been working on a AUV vehicle called Zeno (Figure 1.2). The idea of this vehicle was developed in the framework of the European project for the archeology, 'ARCHEOSUb' [6]. Thanks to its "manta" shape and the actuation of all the 6 Degrees Of Freedom (DOF), it has an high maneuverability and it is able to perform trajectories that a typical torpedo shaped AUV can't do. The quite small size of Zeno makes it a "two-man portable device" and its battery can be changed easily and quickly, also during a sea trial. For all this reasons this vehicle could lead to innovative results, above all in all the tasks that requires high maneuverability. For this reason is fundamental to optimize its motion control.

1.2 Motivation

One of the main areas of interest in the development of an AUV is the trajectory planning. Its goals is to generate the reference inputs to the motion control system which ensure that the vehicle executes the planned trajectory [7]. In order to achieve good results, a suitable motion control system has to be designed. A typical solution is a PID (Proportional-Integral-Derivative) controller, that in general assures stability and rejection to disturbances. Anyway, this kind of controller leads to low performances and high stress on the actuators, due to the fact that the control system is "blind", having no information about the dynamic of the vehicle. For this

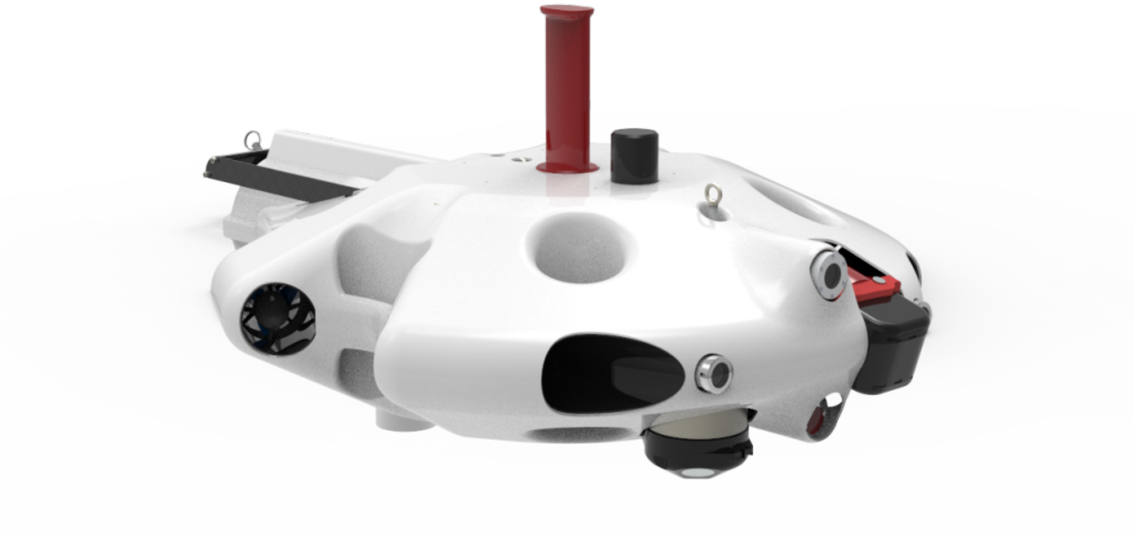


Figure 1.2: Zeno Hull

reason the Team decided to design a controller based on the control law described in (1.1), which put together model-based feedforward terms with PID terms.

$$\begin{aligned}\boldsymbol{\tau}_c =&\hat{\boldsymbol{M}}\dot{\boldsymbol{\nu}}_d + \hat{\boldsymbol{C}}(\boldsymbol{\nu})\boldsymbol{\nu} + \hat{\boldsymbol{D}}(\boldsymbol{\nu}_d)\boldsymbol{\nu}_d + \hat{\boldsymbol{g}}(\boldsymbol{\eta}) + \\ &\hat{\boldsymbol{M}}(\boldsymbol{K_d}(\boldsymbol{\nu}_d - \boldsymbol{\nu}) + \boldsymbol{K_p}(\boldsymbol{\eta}_d - \boldsymbol{\eta}) + \boldsymbol{K_i}\int(\boldsymbol{\eta}_d - \boldsymbol{\eta})\,dt)\end{aligned} \tag{1.1}$$

where:

- $\boldsymbol{\tau}_c$ is the vector of control forces and moments in the body reference frame
- $\hat{\boldsymbol{M}}$ is the estimation of the inertia matrix
- $\hat{\boldsymbol{C}}$ is the estimation of Coriolis and centripetal terms
- $\hat{\boldsymbol{D}}$ is the estimation of the drag matrix
- $\hat{\boldsymbol{g}}$ is the estimation of gravitational force and moments
- $\dot{\boldsymbol{\nu}}_d$ and $\boldsymbol{\nu}_d$ are the desired acceleration and velocity vector in the body frame
- $\dot{\boldsymbol{\nu}}$ and $\boldsymbol{\nu}$ are the measured acceleration and velocity vector in the body frame

- $\boldsymbol{\eta}_d$ and $\boldsymbol{\eta}$ are the desired and measured position and orientation with respect to an inertial reference frame (in nautical application it's common to use a NED reference frame, it will be better shown in section 1.3)

- $\boldsymbol{K_d}$, $\boldsymbol{K_p}$ and $\boldsymbol{K_i}$ are the gain matrices of the PID controller. They have to be symmetric and positive definite.

It can be proved that this controller, properly tuned, ensures the stability of the system. A block diagram of the controller will be shown in section 5.1.

The fundamental advantage of using a feedforward term in the control law is to shift the problem from the vehicle dynamics to the only dynamics error. Indeed, the feedforward term's action tries to compensate the non linear terms due to inertial, Coriolis, centrifugal, gravitational and damping forces, relieving the tracking error rejection task for the PID feedback control action. Hence, this controller allows both better trajectory performance and less effort for the actuators.
On the other hand, the computational effort increases due to the on-line computation of part of the feedforward action. Anyway, this is not a problem in the case of Zeno AUV since its working rate is slow enough (10 Hz). Finally, the goodness of the dynamics model estimation is fundamental for both performance and stability of the vehicle motion. Indeed, using a bad model inside a Model-Based controller could lead to high oscillation or even to instability in the motion. For this reason, the implementation of an identification algorithm for the model parameters it's crucial and this is the starting point of this work.

1.3 Literature Review

The first step in order to create a suitable algorithm for the system identification, is the study of the physical laws behind the system itself. In this particular case, it involves the study of the dynamics of a marine vehicle. This branch of physic is divided in two part: kinematics, that is the study of the geometrical aspect of a motion and kinetics, that instead studies the forces that cause the motion. All the theoretical considerations made in this chapter are based on the contents of the manual [8], written by Fossen.

1.3.1 Kinematics

In order to study the kinematics of a 6 DOF vehicle, firstly it's necessary to define two coordinate frame. A so called "body-fixed reference frame" $X_0Y_0Z_0$ is fixed to the body of the vehicle. The origin O_0 is in general chosen to coincide with the center of gravity (CG) and the axis $X_0Y_0Z_0$ coincide with the principal of inertia. The reference frame and translational and rotational directions are shown in figure 1.3:

- A motion along X_0 direction is called Surge
- A motion along Y_0 direction is called Sway
- A motion along Z_0 direction is called Heave
- A rotation around X_0 direction is called Roll
- A rotation around Y_0 direction is called Pitch
- A rotation around Z_0 direction is called Yaw

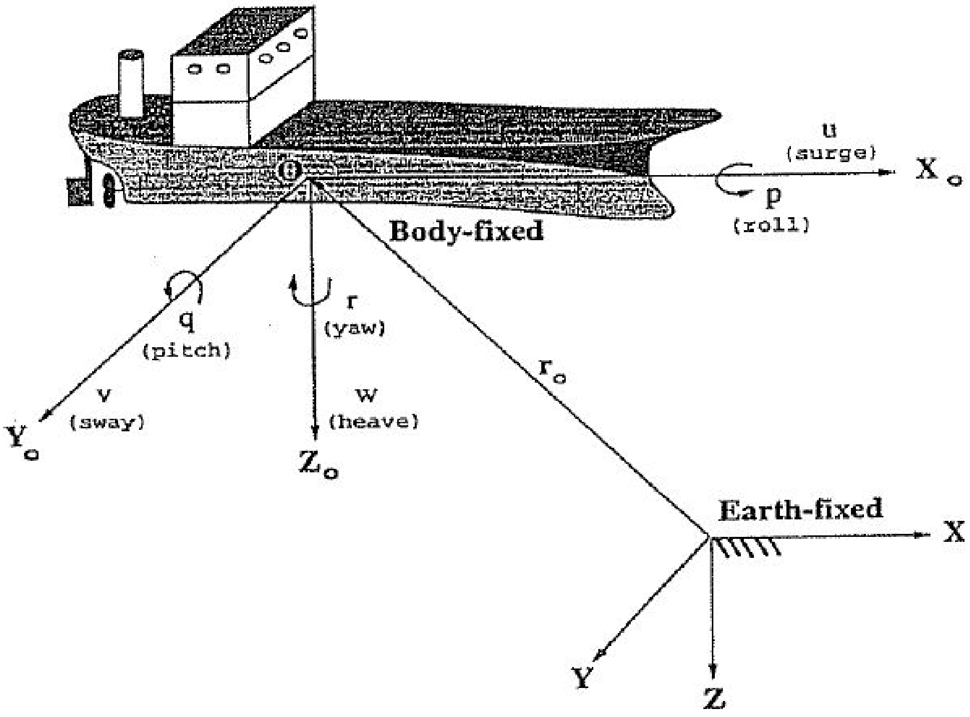


Figure 1.3: NED and Body-Fixed Reference Frames

An inertial reference frame is then necessary for the description of the body-fixed one. In nautical applications, Earth acceleration is negligible because it doesn't affect low speed vehicles like most of the robots used in marine application; for this reason a Earth-fixed reference frame can be considered inertial. In most of the cases it is used a NED reference frame (Nord-East-Down), where the X axis is directed from South to Nord, the Y from West to East, the Z axis from the surface to the center of the Earth. The NED reference frame, has the origin O correspondent to a specific geodetic position, obtained by combining Latitude, Longitude and Altitude. Using NED frame it is possible to establish distances in the cartesian domain rather than in the geodetic space.

The kinematics variables are defined according to the following notation:

$$\begin{array}{lll}
\boldsymbol{\eta} = [\boldsymbol{\eta}_1^T, \boldsymbol{\eta}_2^T]^T; & \boldsymbol{\eta}_1 = [n, e, d]^T; & \boldsymbol{\eta}_2 = [\phi, \theta, \psi]^T; \\
\boldsymbol{\nu} = [\boldsymbol{\nu}_1^T, \boldsymbol{\nu}_2^T]^T; & \boldsymbol{\nu}_1 = [u, v, w]^T; & \boldsymbol{\nu}_2 = [p, q, r]^T; \\
\boldsymbol{\tau} = [\boldsymbol{\tau}_1^T, \boldsymbol{\tau}_2^T]^T; & \boldsymbol{\tau}_1 = [X, Y, Z]^T; & \boldsymbol{\tau}_2 = [K, M, N]^T;
\end{array}$$

Where $\boldsymbol{\eta}$ is the vector of positions and orientations with coordinates described in the inertial reference frame, $\boldsymbol{\nu}$ is the vector of linear and angular velocities in the body-fixed reference frame and $\boldsymbol{\tau}$ is the vector of forces and moments (also named "wrench" or "generalized force") acting on the vehicle in the body-fixed reference frame.
The orientation can be described in marine application by means of Roll-Pitch-Yaw Euler angles or "quaternions".

Euler Angles

The kinematics can be described by the following equation:

$$\dot{\boldsymbol{\eta}} = \boldsymbol{J}(\boldsymbol{\eta})\boldsymbol{\nu} \quad \Longleftrightarrow \quad \begin{bmatrix} \dot{\boldsymbol{\eta}}_1 \\ \dot{\boldsymbol{\eta}}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{J}_1(\boldsymbol{\eta}_2) & \mathbf{0}_{3X3} \\ \mathbf{0}_{3X3} & \boldsymbol{J}_2(\boldsymbol{\eta}_2) \end{bmatrix} \begin{bmatrix} \boldsymbol{\nu}_1 \\ \boldsymbol{\nu}_2 \end{bmatrix} \tag{1.2}$$

where $\dot{\boldsymbol{\eta}}$ is the velocity vector described in the inertial reference frame and $\boldsymbol{J(\eta)}$ is a 6x6 matrix that will be studies better later. Given the block diagonal form of the matrix $\boldsymbol{J(\eta)}$, the equations of linear ad angular kinematics are decoupled: indeed, each motion can be seen as a composition of two elemental motion, a pure translation and a pure rotation. For this reason, also their time evolution can be studied separately.

Linear Velocity Transformation (Euler Angles)

The velocities along the XYZ axis of an Earth-Fixed reference frame can be obtained trough the following transformation:

$$\dot{\boldsymbol{\eta}}_1 = \boldsymbol{J}_1(\boldsymbol{\eta}_2)\boldsymbol{\nu}_1 \tag{1.3}$$

where $\boldsymbol{J}_1(\boldsymbol{\eta}_2)$ is a rotation matrix, function of the Euler angles ϕ, θ, ψ. This is a consequence of the theory of rotations: it can be derived that a rotation matrix describe the change in the relative orientation of two rigid bodies or two vectors. The following two properties will be used:

Property 1.1 (Elementary Rotation Composition)**.** *Each rotation matrix $\boldsymbol{C}$ can be built by the composition of three elementary rotation matrices:*

$$\boldsymbol{C} = \boldsymbol{C}_{v_1,\alpha_1}\boldsymbol{C}_{v_2,\alpha_2}\boldsymbol{C}_{v_3,\alpha_3} \tag{1.4}$$

where $\boldsymbol{C}_{v_i,\alpha_i}$ is a rotation around the reference frame axis $v_i \in \{X, Y, Z\}$ of an angle α_i. In details we have:

$$\boldsymbol{C}_{x,\phi} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\phi & -s\phi \\ 0 & s\phi & c\phi \end{bmatrix} \quad \boldsymbol{C}_{y,\theta} = \begin{bmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{bmatrix} \quad \boldsymbol{C}_{z,\psi} = \begin{bmatrix} c\psi & -s\psi & 0 \\ s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{1.5}$$

where $c\,\cdot = cos(\cdot)$ and $s\,\cdot = sin(\cdot)$

Property 1.2 (Orthonormality)**.** *A rotation matrix C satisfies:*

$$\boldsymbol{C}\boldsymbol{C}^T = \boldsymbol{C}^T\boldsymbol{C} = \boldsymbol{I}; \quad det\boldsymbol{C} = 1 \tag{1.6}$$

As a consequence of this we have that $\boldsymbol{C}^{-1} = \boldsymbol{C}^T$

As mentioned before, in marine application $\boldsymbol{J}_1(\boldsymbol{\eta}_2)$ is built considering the Euler angle of roll pitch and yaw. In particular, from Property (1.1) it is customary to consider a sequence of three elementary rotations, the first about the Z axis (yaw), the second about the Y axis (pitch) the third about the X axis (roll). The rotations are computed around the axes of the rotating reference frame (Intrinsic rotation);

this imply that the composition (product) of the rotation matrix will be done from "left" to "right":

$$\boldsymbol{J}_1(\boldsymbol{\eta}_2) = \boldsymbol{C}_{z,\psi}\boldsymbol{C}_{y,\theta}\boldsymbol{C}_{x,\phi} \tag{1.7}$$

Expanding (1.7) yields:

$$\boldsymbol{J}_1(\boldsymbol{\eta}_2) = \begin{bmatrix} c\psi c\theta & -s\psi c\theta + c\psi s\theta s\phi & s\psi s\phi + c\psi c\phi s\theta \\ s\psi c\theta & c\psi c\phi + s\phi s\theta s\psi & -c\psi s\phi + s\theta s\psi c\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix} \tag{1.8}$$

The inverse transformation is obtained considering Property (1.2):

$$\boldsymbol{\nu}_1 = \boldsymbol{J}_1^T(\boldsymbol{\eta}_2)\dot{\boldsymbol{\eta}}_1 \tag{1.9}$$

Angular Velocity Transformation (Euler Angles)

In order to compute the kinematics of a vehicle it is necessary to compute $\dot{\boldsymbol{\eta}}_2 = [\dot{\phi}, \dot{\theta}, \dot{\psi}]^T$. It possible to obtain this through the following transformation:

$$\dot{\boldsymbol{\eta}}_2 = \boldsymbol{J}_2(\boldsymbol{\eta}_2)\boldsymbol{\nu}_2 \tag{1.10}$$

We can obtain the explicit form of $\boldsymbol{J}_2(\boldsymbol{\eta}_2)$ considering the sequence of rotation that leads to the body-fixed reference frame. First, a rotation is computed around the Z axis of the fixed reference frame. The angular rotation resulting from this rotation represented in the starting reference frame is the vector $\boldsymbol{\omega}_{\psi I} = [0, 0, \dot{\psi}]^T$. Than a second rotation is computed around the Y axis of the new reference frame. The angular rotation represented in the current reference frame is $\boldsymbol{\omega}_{\psi 1} = [0, \dot{\theta}, 0]^T$. Finally the body reference frame is obtained with a rotation around the new X axis, with a resulting angular velocity $\boldsymbol{\omega}_{\psi 2} = [\dot{\phi}, 0, 0]^T$. The sum of these three angular velocities, expressed in the body-fixed reference frame, gives out the expression for $\boldsymbol{\nu}_2$:

$$\boldsymbol{\nu}_2 = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + \boldsymbol{C}_{x,\phi}^T \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \boldsymbol{C}_{x,\phi}^T\boldsymbol{C}_{y,\theta}^T \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} = \boldsymbol{J}_2(\boldsymbol{\eta}_2)^{-1}\dot{\boldsymbol{\eta}}_2 \tag{1.11}$$

Expanding (1.11) yields:

$$\boldsymbol{J}_2(\boldsymbol{\eta}_2)^{-1} = \begin{bmatrix} 1 & 0 & -s\theta \\ 0 & c\phi & c\theta s\phi \\ 0 & -s\phi & c\theta c\phi \end{bmatrix} \implies \boldsymbol{J}_2(\boldsymbol{\eta}_2) = \begin{bmatrix} 1 & s\phi t\theta & c\phi t\theta \\ 0 & c\phi & -s\phi \\ 0 & s\phi/c\theta & c\phi/c\theta \end{bmatrix} \tag{1.12}$$

where where t $\cdot = \tan(\cdot)$.

It can be noticed that the matrix $\boldsymbol{J}_2(\boldsymbol{\eta}_2)$ is not defined for a pitch angle of $\theta = \pm\, \pi/2$. This could lead to problems in the computation of the kinematics when the vehicle is operating near to that condition of singularity. In order to avoid this problem it is possible to use quaternions for the representation of the orientation.

Euler Parameters (Quaternions representation)

As the first step in order to redefine the representation of the orientation of a rigid body, it is necessary to state the following theorem:

Theorem 1.3 (Euler's Theorem on Rotation)**.** *Every change in the relative orientation of two reference frames A and B can be produced by means of a simple rotation of B in A*

In other words, each rotation can be described by means of 4 parameters: a unit vector $\boldsymbol{\lambda}$ (three elements) describing the direction of the rotation axis and an angle β describing the amplitude of the rotation.
Starting from the theorem (1.3) it is possible to define the "Euler parameters quaternion" as

$$\boldsymbol{e} = \begin{bmatrix} \eta \\ \boldsymbol{\epsilon} \end{bmatrix} = \begin{bmatrix} cos(\frac{\beta}{2}) \\ \boldsymbol{\lambda} sin(\frac{\beta}{2}) \end{bmatrix} \tag{1.13}$$

The 4 elements of the vector $\boldsymbol{e}$ are called Euler Parameters and they form a unit quaternion.
It is important to stress the property of "unitarity" of the Euler parameters vector, i.e. $\epsilon_1^2 + \epsilon_2^2 + \epsilon_3^2 + \eta^2 = 1$, because it shows that only 3 of the 4 elements are independent. For this reason the Euler parameter description of the orientation is not minimal.
It's possible to redefine the Kinematic equations using this new representation:

$$\dot{\boldsymbol{\eta}}_{\boldsymbol{E}} = \boldsymbol{E}(\boldsymbol{e})\boldsymbol{\nu} \quad \Longleftrightarrow \quad \begin{bmatrix} \dot{\boldsymbol{\eta}}_1 \\ \dot{\boldsymbol{e}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{E}_1(\boldsymbol{e}) & \boldsymbol{0}_{3X3} \\ \boldsymbol{0}_{3X3} & \boldsymbol{E}_2(\boldsymbol{e}) \end{bmatrix} \begin{bmatrix} \boldsymbol{\nu}_1 \\ \boldsymbol{\nu}_2 \end{bmatrix} \tag{1.14}$$

where $\eta_E = [x, y, z, \epsilon_1, \epsilon_2, \epsilon_3, \eta]^T$ is the vector of positions and orientations.

Linear Velocity Transformation (Euler Parameters)

The equation (1.3) can be rewritten as follow:

$$\dot{\boldsymbol{\eta}}_1 = \boldsymbol{E}_1(\boldsymbol{e})\boldsymbol{\nu}_1 \tag{1.15}$$

Starting from the theorem (1.3), it can be shown that the expanded form of the matrix $\boldsymbol{E}_1(\boldsymbol{e})$ is:

$$\boldsymbol{E}_1(\boldsymbol{e}) = \begin{bmatrix} 1-2(\epsilon_2^2+\epsilon_3^2) & 2(\epsilon_1\epsilon_2-\epsilon_3\eta) & 2(\epsilon_1\epsilon_3+\epsilon_2\eta) \\ 2(\epsilon_1\epsilon_2+\epsilon_3\eta) & 1-2(\epsilon_1^2+\epsilon_3^2) & 2(\epsilon_2\epsilon_3-\epsilon_1\eta) \\ 2(\epsilon_1\epsilon_3-\epsilon_2\eta) & 2(\epsilon_2\epsilon_3+\epsilon_1\eta) & 1-2(\epsilon_1^2+\epsilon_2^2) \end{bmatrix} \tag{1.16}$$

As for the Euler angle representation, being $\boldsymbol{E}_1(\boldsymbol{e})$ a rotation matrix, property (1.2) is still valid, so the inverse transformation $\boldsymbol{E}_1(\boldsymbol{e})^{-1} = \boldsymbol{E}_1(\boldsymbol{e})^T$

Angular Velocity Transformation (Euler Parameters)

Defining the quaternion associated with the angular velocity described as $\boldsymbol{\nu}_2^q = [0, \boldsymbol{\nu}_2]^T$, it can be shown that the time evolution of the Euler parameters quaternion can be calculated as:

$$\dot{\boldsymbol{e}} = \frac{1}{2}\boldsymbol{e} \bigotimes \boldsymbol{\nu}_2^q \tag{1.17}$$

where $\bigotimes$ is the quaternion (or Hamilton) product. Rewriting this equation in matrix form, it is possible to derive the expression desired:

$$\dot{\boldsymbol{e}} = \boldsymbol{E}_2(\boldsymbol{e})\boldsymbol{\nu}_2 \tag{1.18}$$

where

$$\boldsymbol{E}_2(\boldsymbol{e}) = \frac{1}{2} \begin{bmatrix} -\epsilon_1 & -\epsilon_2 & -\epsilon_1 \\ \eta & -\epsilon_3 & \epsilon_2 \\ \epsilon_3 & \eta & -\epsilon_1 \\ -\epsilon_2 & \epsilon_1 & \eta \end{bmatrix} \tag{1.19}$$

Either Euler angles or quaternions can be used in order to represent the attitude of a vehicle. The first representation is more intuitive, but present a singularity and require a large number of trigonometric calculation, which can be heavy from a computational point of view. The second approach has no singularities thanks to the fact that it is a non-minimal representation. Furthermore the computational cost is lower in general. Another problem of the Euler angles representation is the so-called "wraparound" problem, as for example the eventuality of having Euler angles up to values outside the definition interval of ± $\pi/2$ for the pitch and ± π for the roll and the yaw. This problem is not present in the quaternion representation.

1.3.2 Dynamics

This section will first present the general model that describes the dynamics of a vehicle. In the second part a suitable modification to this model will be introduced in order to consider the hydrodynamic contribute in marine applications.

Dynamic model derivation for Rigid-Body

Let's consider a rigid body in $\mathbb{R}^3$ (the ensemble of real numbers in dimension 3 space) subject to external forces $\boldsymbol{F}^{tot}$ and moments $\boldsymbol{M}^{tot}$, an earth-fixed reference frame XYZ and a body-fixed reference frame $X_0Y_0Z_0$ with origin in O. The position of O and the direction of the axes is arbitrary but it is typical to choose O coincident with the center of gravity C_g and/or the $X_0Y_0Z_0$ directions coincident with the symmetry axes of the vehicle. Sometimes it is not possible to make both these choices because the intersection of the symmetry axes does not necessarily coincide with the center of mass. For this reason the following formulation will consider O and C_g as two distinct points.

The dynamics of the body is described by the following two laws, called Euler or Cardinal Equations:

$$\dot{\boldsymbol{p}}_C = \boldsymbol{F}_C^{tot} \qquad \boldsymbol{p}_C := m\boldsymbol{v}_C \tag{1.20}$$

$$\dot{\boldsymbol{h}}_O = \boldsymbol{M}_O^{tot} - \boldsymbol{v}_O \times \boldsymbol{p}_C \qquad \boldsymbol{h}_O := \boldsymbol{I}_O\boldsymbol{\omega} \tag{1.21}$$

The first equation describes the effect of the external forces $\boldsymbol{F}_C^{tot}$ on the linear motion of the body, where $\boldsymbol{p}_C$ is the linear momentum and m is the total mass of the

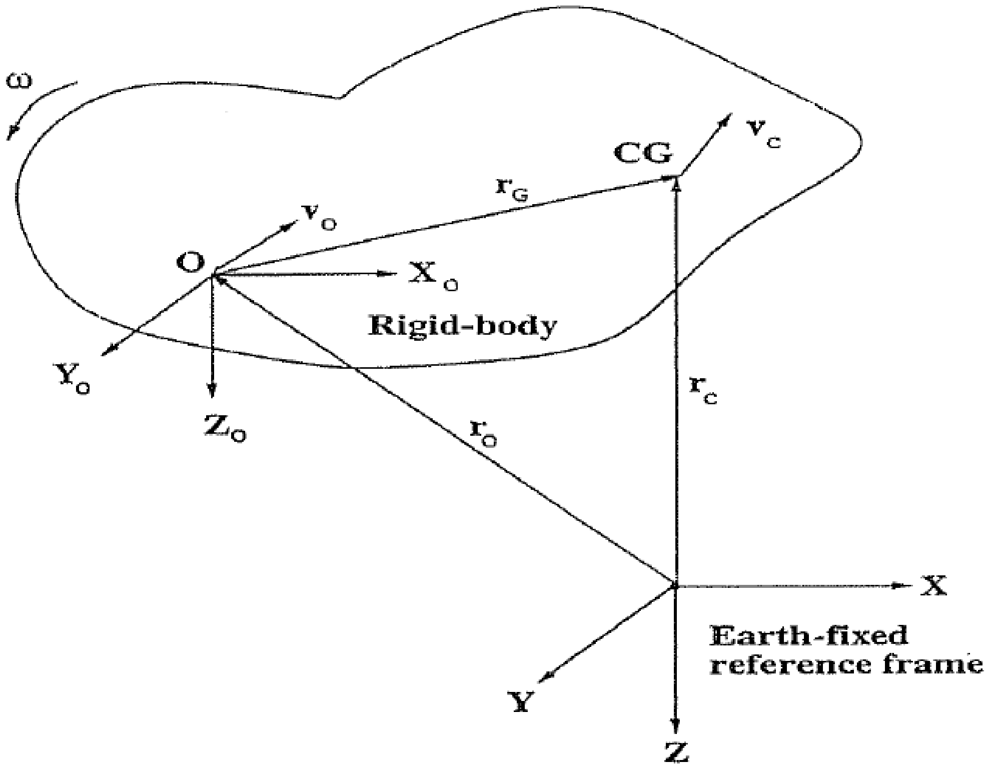


Figure 1.4: Inertial and moving reference frames for the dynamic description of a rigid-body

body. The subscript "$_C$" indicates that all the quantities are referred to the body's center of gravity.
The second equation instead describes the effect of the external moments $\boldsymbol{M}_O^{tot}$ on the angular motion of the body, where $\boldsymbol{h}_O$ is the angular momentum, $\boldsymbol{I}_O$ is the inertia tensor and $\boldsymbol{v}_O$ is the velocity of the origin O and $\boldsymbol{\omega}$ is the angular velocity of the body. The subscript "$_O$" indicates that all the quantities are referred to the origin of the body-fixed reference frame.
Both the equations are three-dimensional vector equations.

In order to extend the first cardinal equation, it is necessary to calculate the time derivative of the expression of $\boldsymbol{p}_C$. From figure (1.4), $\boldsymbol{r}_C = \boldsymbol{r}_O + \boldsymbol{r}_G$, that are respectively the position of the center of gravity, the position of the origin of the body fixed reference frame and the distance between the two points described in the inertial reference frame. Evaluating the second derivative with respect to time, it follows:

$$\dot{\boldsymbol{v}}_C = \dot{\boldsymbol{v}}_O|_b + \boldsymbol{\omega} \times \boldsymbol{v}_O + \dot{\boldsymbol{\omega}}|_b \times \boldsymbol{r}_G + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \boldsymbol{r}_G) \tag{1.22}$$

Where $\boldsymbol{v}_C$ is the velocity of the center of gravity, $\boldsymbol{v}_O$ is the velocity of the origin of the body-fixed reference frame. The symbol $|_b$ indicates that the time derivative is calculated in the moving reference frame. In order to obtain the equation (1.22) it is necessary to utilize the formula $\dot{\boldsymbol{c}} = \dot{\boldsymbol{c}}|_b + \boldsymbol{\omega} \times \boldsymbol{c}$ which puts in relationship the derivative of a vector with respect to both the reference frames considered.

Substituting (1.22) into (1.20) we obtain:

$$\boldsymbol{F}_C^{tot} = m\dot{\boldsymbol{v}}_C = m(\dot{\boldsymbol{v}}_O|_b + \boldsymbol{\omega} \times \boldsymbol{v}_O + \dot{\boldsymbol{\omega}}|_b \times \boldsymbol{r}_G + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \boldsymbol{r}_G)) \tag{1.23}$$

In the case of O coinciding with the center of gravity, the distance $\boldsymbol{r}_G = 0$; as a consequence this equation has a simpler form:

$$\boldsymbol{F}_C^{tot} = m(\dot{\boldsymbol{v}}_C|_b + \boldsymbol{\omega} \times \boldsymbol{v}_C) \tag{1.24}$$

For what concern the second cardinal equation, it is necessary to find the expression of $\boldsymbol{h}_O$. It's necessary to start defining:

$$\boldsymbol{h}_O := \int_V \boldsymbol{r} \times \boldsymbol{v}\rho_A \, dV \tag{1.25}$$

$$\boldsymbol{I}_O := -\int_V \boldsymbol{S}(\boldsymbol{r})\boldsymbol{S}(\boldsymbol{r})\rho_A \, dV \tag{1.26}$$

where

$$\boldsymbol{S}(\boldsymbol{r}) := \begin{bmatrix} 0 & -r_3 & r_2 \\ r_3 & 0 & -r_1 \\ -r_2 & r_1 & 0 \end{bmatrix} \tag{1.27}$$

is the skew-symmetrical matrix associated with the vector $\boldsymbol{r}$. This representation is also connected with the vector product, having $\boldsymbol{S}(\boldsymbol{r}) = \boldsymbol{r}\times$.

Now, remembering that $\boldsymbol{v} = \boldsymbol{v}_O + \boldsymbol{\omega} \times \boldsymbol{r}$, it is possible to write (1.25) as

$$\boldsymbol{h}_O = \int_V \boldsymbol{r} \times \boldsymbol{v}_O\rho_A \, dV + \int_V \boldsymbol{r} \times (\boldsymbol{\omega} \times \boldsymbol{r})\rho_A \, dV \tag{1.28}$$

The first term can be rewritten as

$$\int_V \boldsymbol{r} \times \boldsymbol{v}_O \rho_A \, dV = (\int_V \boldsymbol{r} \rho_A \, dV) \times \boldsymbol{v}_O = m\boldsymbol{r}_G \times \boldsymbol{v}_O$$

using the definition of center of gravity

$$\boldsymbol{r}_G := \frac{1}{m} \int_V \boldsymbol{r} \rho_A \, dV \tag{1.29}$$

The second term can be rewritten using (1.26), after some calculation:

$$\begin{aligned}&\int_V \boldsymbol{r} \times (\boldsymbol{\omega} \times \boldsymbol{r}) \rho_A \, dV = -\int_V \boldsymbol{r} \times (\boldsymbol{r} \times \boldsymbol{\omega}) \rho_A \, dV = \\ &= (-\int_V \boldsymbol{S}(\boldsymbol{r}) \boldsymbol{S}(\boldsymbol{r}) \rho_A \, dV) \boldsymbol{\omega} = \boldsymbol{I}_O \boldsymbol{\omega}\end{aligned}$$

Substituting these two results above inside the equation (1.28), it reduces to:

$$\boldsymbol{h}_O = \boldsymbol{I}_O \boldsymbol{\omega} + m\boldsymbol{r}_G \times \boldsymbol{v}_O \tag{1.30}$$

Computing the time derivative of this expression and substituting it into (1.21) the final result is:

$$\boldsymbol{M}_O^{tot} = \dot{\boldsymbol{h}}_O + \boldsymbol{v}_0 \times \boldsymbol{p} = \boldsymbol{I}_O \dot{\boldsymbol{\omega}}|_b + \boldsymbol{\omega} \times (\boldsymbol{I}_O \boldsymbol{\omega}) + m\boldsymbol{r}_G \times (\dot{\boldsymbol{v}}_O|_b + \boldsymbol{\omega} \times \boldsymbol{v}_O) \tag{1.31}$$

having $m(\boldsymbol{\omega} \times \boldsymbol{r}_G) \times \boldsymbol{v}_O$ simplified with $\boldsymbol{v}_0 \times \boldsymbol{p} = m\boldsymbol{v}_0 \times (\boldsymbol{\omega} \times \boldsymbol{r}_G)$. As before, if O coincides with the center of gravity, the equation is reduced to a simpler form:

$$\boldsymbol{M}_C^{tot} = \boldsymbol{I}_C \dot{\boldsymbol{\omega}}|_b + \boldsymbol{\omega} \times (\boldsymbol{I}_C \boldsymbol{\omega}) \tag{1.32}$$

Vectorial Formulation of 6 DOF Rigid-Body Equation of Motion

Linear equation (1.23) and angular equation (1.31) can be represented together in a compact form, which we will refer with subscript "$_{RB}$":

$$\boldsymbol{M}_{RB} \dot{\boldsymbol{\nu}} + \boldsymbol{C}_{RB}(\boldsymbol{\nu}) \boldsymbol{\nu} = \boldsymbol{\tau}_{RB} \tag{1.33}$$

where $\boldsymbol{\nu}$ is the body-fixed linear and angular velocity vector and $\boldsymbol{\tau}_{RB} = [\boldsymbol{F}_C^{tot}, \boldsymbol{M}_O^{tot}]^T = [X, Y, Z, K, M, N]^T$ is the generalized vector of external forces and moments, $\boldsymbol{M}_{RB}$ is the inertia matrix and $\boldsymbol{C}_{RB}$ is the coriolis-centripetal matrix..

Property 1.4 ($\boldsymbol{M}_{RB}$)**.** *The parameterization of $\boldsymbol{M}_{RB}$ is unique and positive definite:*

$$\boldsymbol{M}_{RB} = \boldsymbol{M}_{RB}^T > \boldsymbol{0}$$

Where

$$\boldsymbol{M}_{RB} = \begin{bmatrix} m\boldsymbol{I}_{3\times3} & -m\boldsymbol{S}(\boldsymbol{r}_G) \\ m\boldsymbol{S}(\boldsymbol{r}_G) & \boldsymbol{I}_O \end{bmatrix} \tag{1.34}$$

with $\boldsymbol{I}_{3\times3}$ identity matrix and $\boldsymbol{I}_O$ positive definite too. Moreover $\dot{\boldsymbol{M}}_{RB}$=0, so it does not depend on time.

Theorem 1.5 ($\boldsymbol{C}_{RB}$)**.** *Defining the inertia matrix as:*

$$\boldsymbol{M}_{RB} = \begin{bmatrix} \boldsymbol{M}_{11} & \boldsymbol{M}_{12} \\ \boldsymbol{M}_{21} & \boldsymbol{M}_{22} \end{bmatrix}$$

it is always possible to parameterize the Coriolis and centripetal matrix so that $\boldsymbol{C}_{RB} = -\boldsymbol{C}_{RB}^T$, defined as follow:

$$\boldsymbol{C}_{RB} = \begin{bmatrix} \boldsymbol{0}_{3\times3} & -\boldsymbol{S}(\boldsymbol{M}_{11}\boldsymbol{\nu}_1 + \boldsymbol{M}_{12}\boldsymbol{\nu}_2) \\ -\boldsymbol{S}(\boldsymbol{M}_{11}\boldsymbol{\nu}_1 + \boldsymbol{M}_{12}\boldsymbol{\nu}_2) & -\boldsymbol{S}(\boldsymbol{M}_{21}\boldsymbol{\nu}_1 + \boldsymbol{M}_{22}\boldsymbol{\nu}_2) \end{bmatrix} \tag{1.35}$$

Differently from $\boldsymbol{M}_{RB}$, the parameterization of the Coriolis and centripetal matrix is not unique. Another useful representation for $\boldsymbol{C}_{RB}$ is the following:

$$\boldsymbol{C}_{RB} = \begin{bmatrix} m\boldsymbol{S}(\boldsymbol{\nu}_2) & -m\boldsymbol{S}(\boldsymbol{\nu}_2)\boldsymbol{S}(\boldsymbol{r}_G) \\ m\boldsymbol{S}(\boldsymbol{\nu}_2)\boldsymbol{S}(\boldsymbol{r}_G) & -\boldsymbol{S}(\boldsymbol{I}_O\boldsymbol{\nu}_2) \end{bmatrix} \tag{1.36}$$

This representation is Skew-symmetrical as the previous but its peculiarity is that it depends only on $\boldsymbol{\nu}_2$ so any change in the linear velocity will not affect it.

Hydrodynamic Model

In order to consider the hydrodynamic contribute to the equation of motion of a 6 DOF rigid-body, it is typical to linearly add suitable forces and moments:

- The Inertia of the fluid that surrounds the body affects the motion of the body itself, acting like an added mass. It can be modeled as an increment on the inertia matrix and a corresponding increment on Coriolis and centripetal terms:

$$\boldsymbol{\tau}_M = -\boldsymbol{M}_A\dot{\boldsymbol{\nu}} - \boldsymbol{C}_A(\boldsymbol{\nu})\boldsymbol{\nu} \tag{1.37}$$

- The energy carried away by generated surface waves can be modeled as a potential damping. Other effects can be introduced, like the skin friction of the fluid, the wave drift and the damping caused by the vortex shedding. As a result all these effects are considered together in a unique term

$$\boldsymbol{\tau}_D = -\boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu} \tag{1.38}$$

- Gravity and Buoyancy forces, called restoring forces, are modeled like a vector depending only on the asset of the body:

$$\boldsymbol{\tau}_G = -\boldsymbol{g}(\boldsymbol{\eta}) \tag{1.39}$$

Putting together all these contributions inside the equation (1.33) it is possible to obtain:

$$\boldsymbol{M}\dot{\boldsymbol{\nu}} + \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{g}(\boldsymbol{\eta}) = \boldsymbol{\tau} \tag{1.40}$$

where

$$\boldsymbol{M} = \boldsymbol{M}_{RB} + \boldsymbol{M}_A \qquad\qquad \boldsymbol{C} = \boldsymbol{C}_{RB} + \boldsymbol{C}_A$$

The added inertia matrix $\boldsymbol{M}_A$ is a complete 6×6 matrix, defined as follow:

$$\boldsymbol{M}_A = \begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} \end{bmatrix} := - \begin{bmatrix} X_{\dot{u}} & X_{\dot{v}} & X_{\dot{w}} & X_{\dot{p}} & X_{\dot{q}} & X_{\dot{r}} \\ Y_{\dot{u}} & Y_{\dot{v}} & Y_{\dot{w}} & Y_{\dot{p}} & Y_{\dot{q}} & Y_{\dot{r}} \\ Z_{\dot{u}} & Z_{\dot{v}} & Z_{\dot{w}} & Z_{\dot{p}} & Z_{\dot{q}} & Z_{\dot{r}} \\ K_{\dot{u}} & K_{\dot{v}} & K_{\dot{w}} & K_{\dot{p}} & K_{\dot{q}} & K_{\dot{r}} \\ M_{\dot{u}} & M_{\dot{v}} & M_{\dot{w}} & M_{\dot{p}} & M_{\dot{q}} & M_{\dot{r}} \\ N_{\dot{u}} & N_{\dot{v}} & N_{\dot{w}} & N_{\dot{p}} & N_{\dot{q}} & N_{\dot{r}} \end{bmatrix} \tag{1.41}$$

In general the 36 element of this matrix could be all distinct but, under the assumption of ideal fluid, no incident waves and no sea current, it can be proven that $\boldsymbol{M}_A = \boldsymbol{M}_A^T > \boldsymbol{0}$. In non-ideal condition, anyway, numerical results shows that $\boldsymbol{M}_A$ can be considered positive definite with good approximation. As a result, also $\boldsymbol{M} = \boldsymbol{M}_{RB} + \boldsymbol{M}_A$ is positive definite, being the sum of positive definite matrices.

The hydrodynamic Coriolis and centripetal matrix $\boldsymbol{C}_A(\boldsymbol{\nu})$ of a vehicle moving inside an ideal fluid can be always parameterized as follows:

$$\boldsymbol{C}_A(\boldsymbol{\nu}) = \begin{bmatrix} \boldsymbol{0}_{3\times3} & -\boldsymbol{S}(\boldsymbol{A}_{11}\boldsymbol{\nu}_1 + \boldsymbol{A}_{11}\boldsymbol{\nu}_2) \\ -\boldsymbol{S}(\boldsymbol{A}_{11}\boldsymbol{\nu}_1 + \boldsymbol{A}_{11}\boldsymbol{\nu}_2) & -\boldsymbol{S}(\boldsymbol{A}_{21}\boldsymbol{\nu}_1 + \boldsymbol{A}_{22}\boldsymbol{\nu}_2) \end{bmatrix} \tag{1.42}$$

The matrix parameterized in this way is skew-symmetrical, so $\boldsymbol{C}_A(\boldsymbol{\nu}) = -\boldsymbol{C}_A^T(\boldsymbol{\nu})$. As a result, also $\boldsymbol{C} = \boldsymbol{C}_{RB} + \boldsymbol{C}_A$ is skew-symmetrical, being the sum of skew-symmetrical matrices.

The hydrodynamic damping, or simply drag matrix $\boldsymbol{D}(\boldsymbol{\nu})$ is, in general, a non symmetric complete matrix. Since the damping forces are known to be dissipative, $\boldsymbol{D}(\boldsymbol{\nu}) > 0$. In the next chapter it will be shown a simpler form for this matrix, due to the difficulties in the identification of its most general form.

The restoring forces and moments vector $\boldsymbol{g}(\boldsymbol{\eta})$ takes into account the effects of the gravitational force on the center of gravity $\boldsymbol{r}_G$ and of the buoyant force on the center of buoyancy $\boldsymbol{r}_B$. Defining as $[0, 0, W]^T$ the gravitational force and as $[0, 0, B]^T$ the buoyancy force, both in the inertial reference frame (in nautical application the z-axis is taken positive downwards), their representation in the body-fixed reference frame is:

$$\boldsymbol{f}_G(\boldsymbol{\eta}_2) = \boldsymbol{J}_1^{-1}(\boldsymbol{\eta}_2) \begin{bmatrix} 0 \\ 0 \\ W \end{bmatrix} \qquad \boldsymbol{f}_B(\boldsymbol{\eta}_2) = -\boldsymbol{J}_1^{-1}(\boldsymbol{\eta}_2) \begin{bmatrix} 0 \\ 0 \\ B \end{bmatrix}$$

As a consequence, the representation of the restoring vector is:

$$\boldsymbol{g}(\boldsymbol{\eta}) = -\begin{bmatrix} \boldsymbol{f}_G(\boldsymbol{\eta}_2) + \boldsymbol{f}_B(\boldsymbol{\eta}_2) \\ \boldsymbol{r}_G \times \boldsymbol{f}_G(\boldsymbol{\eta}_2) + \boldsymbol{r}_B \times \boldsymbol{f}_B(\boldsymbol{\eta}_2) \end{bmatrix} \tag{1.43}$$

The sign of the vector has been changed, considering that in equation (1.39) it has been defined with the minus sign.

Ocean currents, wind and incident waves can be considered as a unique term, that in general is not easy to be modelled. For this reason it is treated as a disturbance force $\boldsymbol{\tau}_E$ to be estimated. Another way to model the contribute of currents is to consider the relative velocity of the body with respect to the fluid $\boldsymbol{\nu}_r = \boldsymbol{\nu} - \boldsymbol{\nu}_c$, and rewrite the equation of the dynamics in terms of $\boldsymbol{\nu}_r$. If the current is irrotational and constant with respect to NED frame $\boldsymbol{\nu}_c = [u_c, v_c, w_c, 0, 0, 0]^T$ and $\dot{\boldsymbol{\nu}}_c = 0$. It can be proven that the equation of motion can be rewritten as follows:

$$\boldsymbol{M}\dot{\boldsymbol{\nu}}_r + \boldsymbol{C}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r + \boldsymbol{D}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r + \boldsymbol{g}(\boldsymbol{\eta}) = \boldsymbol{\tau} \tag{1.44}$$

with $\boldsymbol{C}_{RB}$ parameterized as in (1.36). So it is shown that, considering a Coriolis and centripetal formulation that doesn't depends on the linear velocity ν_1, the value of parameters for the equation of dynamics does not change if the relative velocity is considered instead of the body-frame velocity.

In conclusion, putting together the equations of kinematics and dynamics, the motion of a generic vehicle inside a fluid can be mathematical described as follows:

$$\begin{gathered} \boxed{\boldsymbol{M}\dot{\boldsymbol{\nu}} + \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{g}(\boldsymbol{\eta}) = \boldsymbol{\tau}} \\ \boxed{\dot{\boldsymbol{\eta}} = \boldsymbol{J}(\boldsymbol{\eta})\boldsymbol{\nu}} \end{gathered} \tag{1.45}$$

Chapter 2

Zeno AUV Model

In this chapter it is shown how to adapt the equation of motion a specific vehicle, in particular the Zeno AUV. For my master thesis, I have worked on the study of the motion of this vehicle, developed by the MDM team.

As stated in the first chapter, Zeno has all the 6 DoF actuated, thanks to its 8 brushless motor propellers. Four of the propellers are located so that they can generate a propulsion normal to the plane which cut the vehicle horizontally, allowing the motion of heave, roll and pitch. The remaining four propellers can generate thrusts acting along the horizontal plane, but with different direction. In this way it is possible to actuate the Sway motion, together with Surge and Yaw. In figure (2.1) it is shown the disposition of the 8 propellers.

The total propulsion granted by the thrusters can be mapped into the vector of forces and moments in the body fixed reference frame thanks to the thruster allocation matrix. I profited from the estimate of the allocation matrix which was already implented in the software of Zeno, that allowed me to consider directly the vector of forces and moments $\boldsymbol{\tau}$.

In the next two sections it is shown how the equations of Kinematics and Dynamics are adapted to the specific case of Zeno.

2.1 Kinematics

The kinematics of Zeno has been implemented using quaternions. This solution has been chosen due to the fact that, having all the DOF actuated, the vehicle could

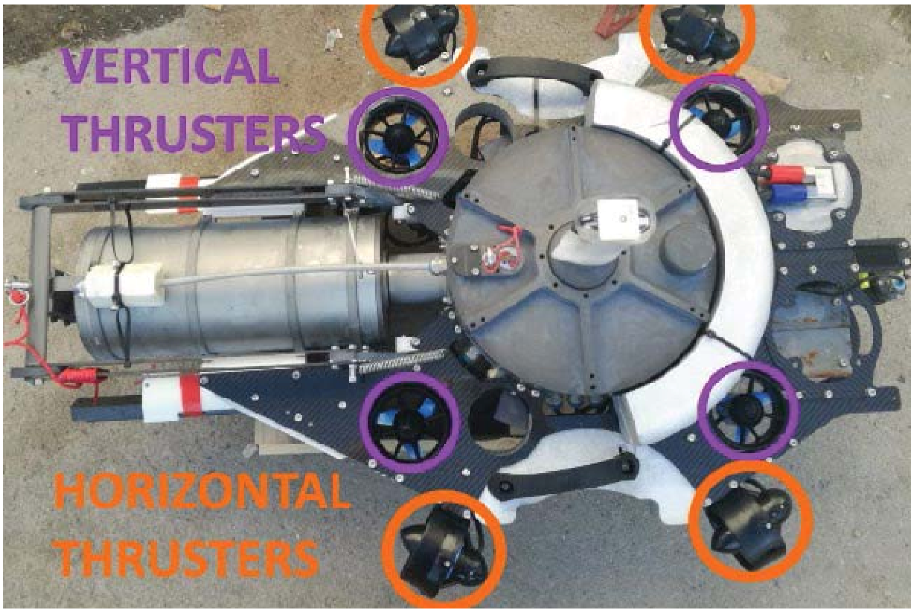


Figure 2.1: Thrusters disposition of Zeno

moves near to the singularity of the Euler angles representation, presented in the first chapter. In second place, even if the identification process has been developed off-line, it is preferable to optimize the computational cost of the algorithm, in order to spend less time during the simulation and validation phase.

In figure (2.2) a block scheme of the kinematic simulator is presented. As it is possible to notice, the output to the user presents the conversion from quaternion to Roll-Pitch-Yaw angles. This last representation indeed has an immediate physical interpretation and correspond to the representation of the real data coming out from the sensors mounted on Zeno.

In order to measure the kinematics quantities, Zeno is provided with different sensors. Two different gyroscopes are mounted on it. One of them is integrated onto an IMU (Inertial Measurement Unit), the second one is a FOG (Fibre-Optic Gyroscope). The outputs of these two sensors are filtered and elaborated in order to provide the angular velocities and, integrating, the Euler angles in Roll, Pitch, Yaw form. Measurement errors are small thanks to the extremely high precision of the FOG. A DVL (Doppler Velocity Log) provides the relative velocity of Zeno with respect to the water, with high precision. This sensor is fundamental for the Odometry of the vehicle during the navigation because under the surface of the water

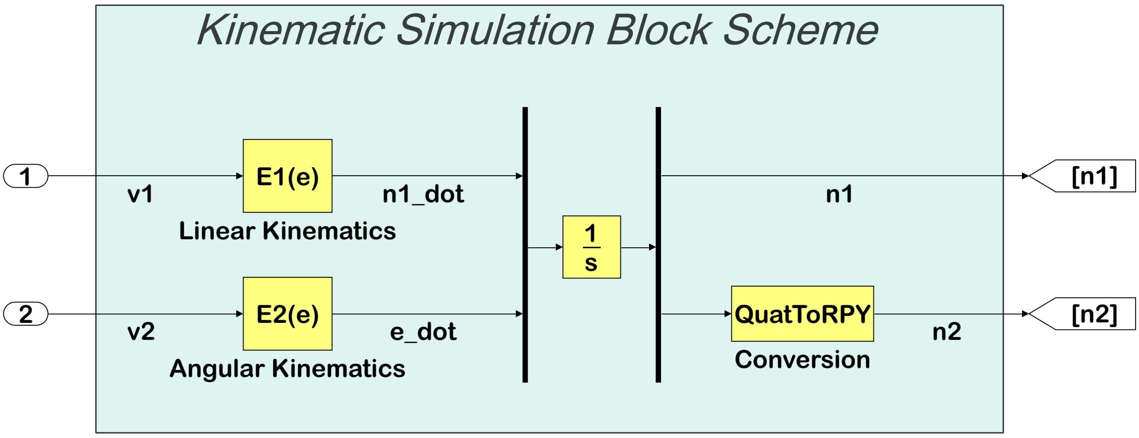


Figure 2.2: Block Diagram of Zeno Kinematics

it is impossible to get information from the GPS. Moreover, three accelerometers are present inside the IMU, for the measurement of the accelerations along each axes of the IMU reference frame. Considering the IMU reference frame coincident with the body fixed reference frame of the vehicle and neglecting the bias and the noise of the sensor, the output is the following:

$$\boldsymbol{a}_{imu} = \boldsymbol{R}^T(\boldsymbol{a}_i - \boldsymbol{g}) \tag{2.1}$$

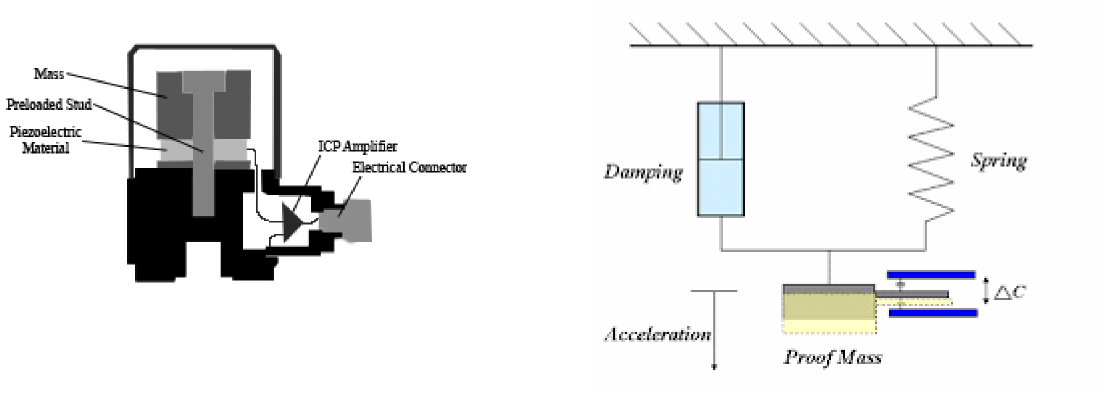
where $\boldsymbol{a}_{imu}$ is the acceleration measured by the sensors along the XYZ body axes, $\boldsymbol{R}$ is the rotation matrix that changes the representation of a vector from the body fixed reference frame to the inertial one, $\boldsymbol{a}_i$ is the acceleration of the body in the NED reference frame and $\boldsymbol{g}$ is the vector of gravity represented in the NED frame too. Considering that the velocity of the vehicle in the inertial reference frame can be written as

$$\boldsymbol{v}_i = \boldsymbol{R}\boldsymbol{v}_b \tag{2.2}$$

Where $\boldsymbol{v}_b$ is the velocity in the body fixed reference frame. Deriving this expression and substituting the result inside the equation (2.1) it's possible to obtain a final expression for the IMU:

$$\boldsymbol{a}_{imu} = \boldsymbol{a}_i + \boldsymbol{S}(\boldsymbol{\omega}_b)\boldsymbol{v}_b - \boldsymbol{R}^T\boldsymbol{g} \tag{2.3}$$

Where $\boldsymbol{S}(\boldsymbol{\omega}_b)$ is the skew-symmetric matrix of the angular velocity vector written in the body reference frame. This expression is used both for simulate the output of a virtual accelerometer and correct the measurements coming from the IMU, before using them as inputs for the identification process. In figure (2.3) it is shown its working principle.



(a) Piezoresistive Accellerometer

(b) Capacitive Accellerometer

Figure 2.3: Working Principle of Two Different Type of Accelerometer

2.2 Dynamics

The Dynamics of Zeno is studied starting from equation (1.40), that is reminded below:

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

First of all, some simplifications can be applied. The origin of the body fixed reference frame has been chosen coincident with the center of gravity; the direction of the XYZ axes has been chosen coincident with the direction of the symmetry axes of the vehicle. The values of the mass and of the body inertia matrix are known from CAD (Computer Aided Design).

$$\boldsymbol{M}_{RB} = \begin{bmatrix} m & 0 & 0 & 0 & 0 & 0 \\ 0 & m & 0 & 0 & 0 & 0 \\ 0 & 0 & m & 0 & 0 & 0 \\ 0 & 0 & 0 & I_x & 0 & 0 \\ 0 & 0 & 0 & 0 & I_y & 0 \\ 0 & 0 & 0 & 0 & 0 & I_z \end{bmatrix} \tag{2.4}$$

where m = 42.56 kg is the mass of Zeno and $[I_x, I_y, I_z] = [0.94, 3.28, 3.79]$ $kg \cdot m^2$ are the diagonal terms of the inertia matrix $\boldsymbol{I}_O$.

The expression has been simplified rounding the values to the second decimal digit and neglecting the off-diagonal terms, smaller of two orders of magnitude with respect to the diagonal ones. This means that the principal axes nearly coincide with the axes of symmetry and so, with the axes of the body reference frame.

The term $\boldsymbol{C}_{RB}$ depends only on the values of the inertia matrix and on the values of the velocity in the Body-Fixed reference frame. For these reasons it can be considered as a known matrix during the identification process. It has been represented using the formulation (1.36), that can be made explicit considering the previous considerations:

$$\boldsymbol{C}_{RB} = \begin{bmatrix} 0 & -mr & mq & 0 & 0 & 0 \\ mr & 0 & -mp & 0 & 0 & 0 \\ -mq & mp & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & I_z r & -I_y q \\ 0 & 0 & 0 & -I_z r & 0 & I_x p \\ 0 & 0 & 0 & I_y q & -I_x p & 0 \end{bmatrix} \tag{2.5}$$

where p,q and r are the angular components of the body fixed velocity vector $\boldsymbol{\nu}$.

In this way the model is valid both for absolute velocity and velocity relative to the current, as mentioned in the previous chapter. This choice has been made considering future applications of the simulator. Indeed, all the identification tests have been performed in a controlled situation, inside a testing pool of the Team's laboratory. For this reason, current velocity and disturbances of the environment can be considered negligible.

For what concern the added mass contribute, as stated in section 1.3, it can be modelled as a complete 6x6 matrix. Anyway, in the case of Zeno, the expression can be simplified considering the symmetries of the vehicle. In fact it presents a port/starboard symmetry (X-Z plane of symmetry) and for the moment, it has been considered a bottom/top symmetry (X-Y plane of symmetry). As a result, the matrix (1.41) can be rewritten in the following simplified form:

$$\boldsymbol{M}_A = \begin{bmatrix} m_{A11} & 0 & 0 & 0 & 0 & 0 \\ 0 & m_{A22} & 0 & 0 & 0 & m_{A26} \\ 0 & 0 & m_{A33} & 0 & m_{A35} & 0 \\ 0 & 0 & 0 & m_{A44} & 0 & 0 \\ 0 & 0 & m_{A53} & 0 & m_{A55} & 0 \\ 0 & m_{A26} & 0 & 0 & 0 & m_{A66} \end{bmatrix} \tag{2.6}$$

With respect to the formulation of the Fossen (1.41), a different notation have been considered. The position of the element inside the matrix is determinated by the two numbers at the subscript The letter "A" stands for "Added". All the values of this matrix have to be identified.

The second condition of symmetry (X-Y plane of symmetry) is non totally satisfied, due to the present of the DVL sensor mounted on the bottom and to the radio-transmission antenna on the top. Anyway, it is always possible to improve the model in a second moment, considering all the terms neglected thanks to this condition. The result will be an increase of the number of variables inside the identification problem that will be presented in the next chapter.

The corresponding hydrodynamic Coriolis and centripetal matrix is obtained using the definition (1.42). This matrix do not present new terms to be identified. Indeed, it depends only on the values of the added mass inertia matrix and of the linear and angular velocities in Body fixed reference frame, that are considered as an input in the identification problem.

The dumping contribution has been taken in account considering two different drag matrices. The first takes into account the friction effects of the water proportional with the velocity (linear contribution), the second is proportional to the square of it (quadratic contribution). Also in this case, they are complete 6x6 matrices. Anyway, as stated from Fossern [8], for a vehicle with three plane of symmetry, moving underwater and performing a non-coupled motion at low speed, it is possible to consider the two matrices symmetric and diagonal. In our case, having only two plane of symmetry, the following formulation have been adopted:

$$\boldsymbol{D}_L = \begin{bmatrix} d_{L11} & 0 & 0 & 0 & 0 & 0 \\ 0 & d_{L22} & 0 & 0 & 0 & d_{L26} \\ 0 & 0 & d_{L33} & 0 & d_{L35} & 0 \\ 0 & 0 & 0 & d_{L44} & 0 & 0 \\ 0 & 0 & d_{L53} & 0 & d_{L55} & 0 \\ 0 & d_{L26} & 0 & 0 & 0 & d_{L66} \end{bmatrix} \tag{2.7}$$

$$\boldsymbol{D}_Q = \begin{bmatrix} d_{Q11}|u| & 0 & 0 & 0 & 0 & 0 \\ 0 & d_{Q22}|v| & 0 & 0 & 0 & d_{Q26}|v| \\ 0 & 0 & d_{Q33}|w| & 0 & d_{Q35}|w| & 0 \\ 0 & 0 & 0 & d_{Q44}|p| & 0 & 0 \\ 0 & 0 & d_{Q53}|q| & 0 & d_{Q55}|q| & 0 \\ 0 & d_{Q26}|r| & 0 & 0 & 0 & d_{Q66}|r| \end{bmatrix} \tag{2.8}$$

where $|.|$ is the absolute value of the corresponding term of the velocity vector $\boldsymbol{\nu}$. d_{Lji} and d_{Qji} have been considered constant. As it is possible to see, the lack of one plane of symmetry leads to the presence of four coupling terms, equal two by two for the symmetry of the matrices. Both the values d_{Lji} and d_{Qji} have to be identified.

The restoring forces and moments vector can be simplified too, considering that $\boldsymbol{r}_G = 0^T$. The position of the center of buoyancy had already been estimated before I started my work.
The value of the buoyancy force could be estimated. Firstly, it is important to state that during the mechanical design, it is fundamental to project a vehicle that is able to float in its rest situation. This choice is made for a security reason; indeed, the vehicle must be anyway able to get back to the surface, even if it is switched off, if any kind of error occurs during the underwater navigation. For this reason a set of buoyancy foams has been applied below the hull. Anyway the difference between the buoyancy and gravitational forces has to be really small otherwise the controller should spend too much energy just to keep the vehicle in a stationary position. This difference could be estimated just considering the mean value of the thrusts generated by the propellers in order to keep the vehicle stationary in a desired underwater position with null roll and pitch angles.
Putting all together, the following vector is obtained:

$$\boldsymbol{g}(\boldsymbol{\eta}) = -\begin{bmatrix} \boldsymbol{f}_G(\boldsymbol{\eta}_2) + \boldsymbol{f}_B(\boldsymbol{\eta}_2) \\ \boldsymbol{r}_B \times \boldsymbol{f}_B(\boldsymbol{\eta}_2) \end{bmatrix} \tag{2.9}$$

where

$$\boldsymbol{f}_G(\boldsymbol{\eta}_2) = \boldsymbol{J}_1^{-1}(\boldsymbol{\eta}_2)\begin{bmatrix}0\\0\\415.92\end{bmatrix}N \qquad \boldsymbol{f}_B(\boldsymbol{\eta}_2) = -\boldsymbol{J}_1^{-1}(\boldsymbol{\eta}_2)\begin{bmatrix}0\\0\\420.92\end{bmatrix}N$$

$$\boldsymbol{r}_B = \begin{bmatrix}0\\0\\-0.006\end{bmatrix}m$$

The values of the angles $\boldsymbol{\eta}_2$ are inputs during the identification process, for this reason the restoring force vector is a known quantity, without introducing new variables.

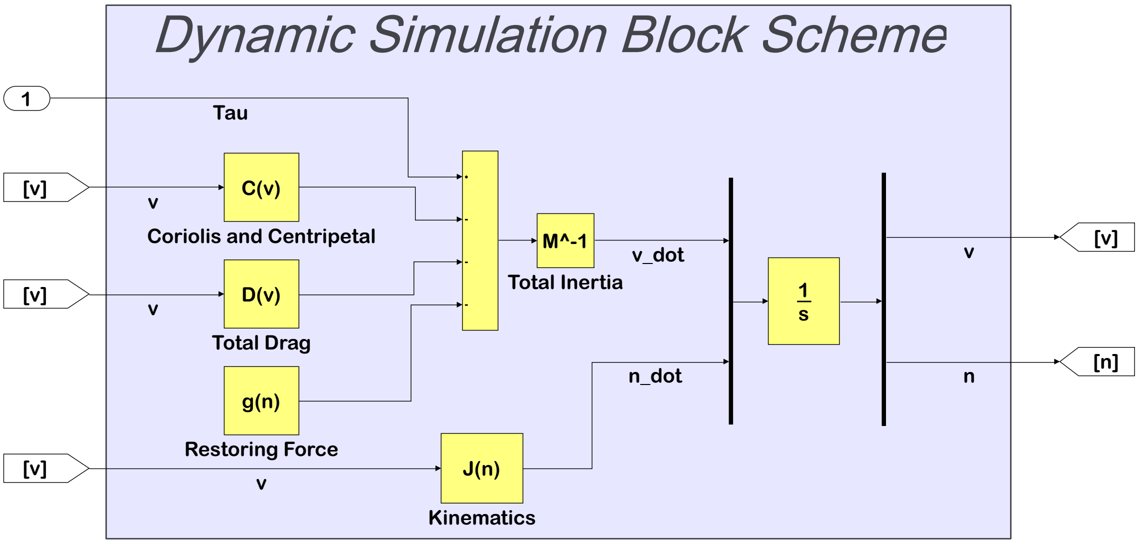


Figure 2.4: Block Diagram of Zeno Dynamics

In figure (2.4) it's presented a block scheme of the dynamics of the vehicle. A simulator has been implemented using the programming language of MATLAB. The simulator takes as input the vector $\boldsymbol{\tau}$ with a rectangular shape. The program implements two virtual sensors corrispondent to the DVL and the IMU mounted on Zeno. A file.Mat is generated, containing the values of the outputs of the virtual sensors at the same sampling rate of the real ones. The real data sets of the tests performed in the testing pool of the laboratory are converted form file.BAG (typical format in ROS implementation) to file.MAT, with the same structure of the data sets generated as outputs of the simulator. In this way the identification process can be performed using both real or virtual data, without any change in the code. Bias and white noise are added to these values, except for the gyroscope, that can be considered ideal, thanks to the high precision of the FOG. The values of bias and noise has been chosen from the datasheets of the real IMU and DVL.

One example of the dynamic simulation is now presented, where the unknown values of the added mass and drag matrix have been chosen arbitrarily, assigning them likely values. In figure (2.5) the input is shown, a rectangular waveform with an amplitude of 80 N on the Z axis. Images (2.6) and (2.7) shows the plots of the data getting out from the virtual IMU, while image (2.8) shows the plots of the ones getting out from the virtual DVL. It is clear that, in this case, the motion along the Z axis is highly coupled with the rotation along the Y axis and with the motion along X axis, even if the coupling off-diagonal terms of mass and drag have been chosen nearly 2 order of magnitude lower than the diagonal terms. This coupling results caused by the hydrodynamics effect leads to difficulties in the identification of all the parameters of Zeno. In the next chapter, it is shown how to deal with this problem.

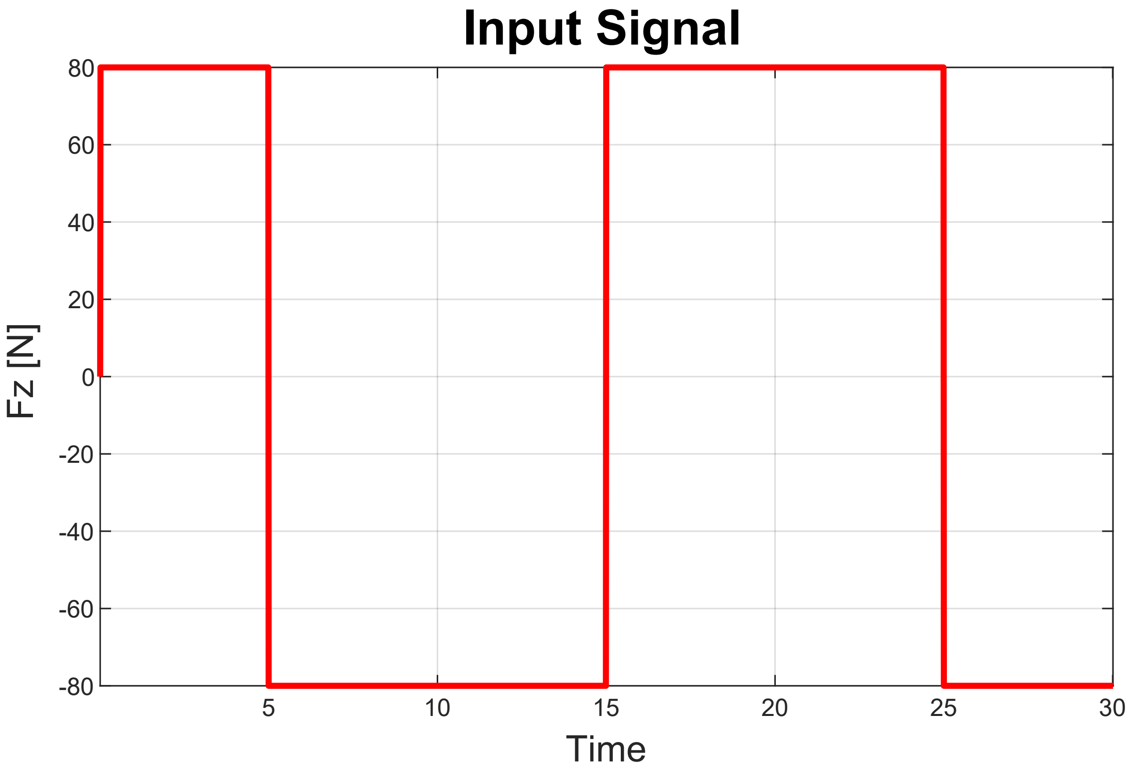


Figure 2.5: Applied Input Force Along the Heave Direction

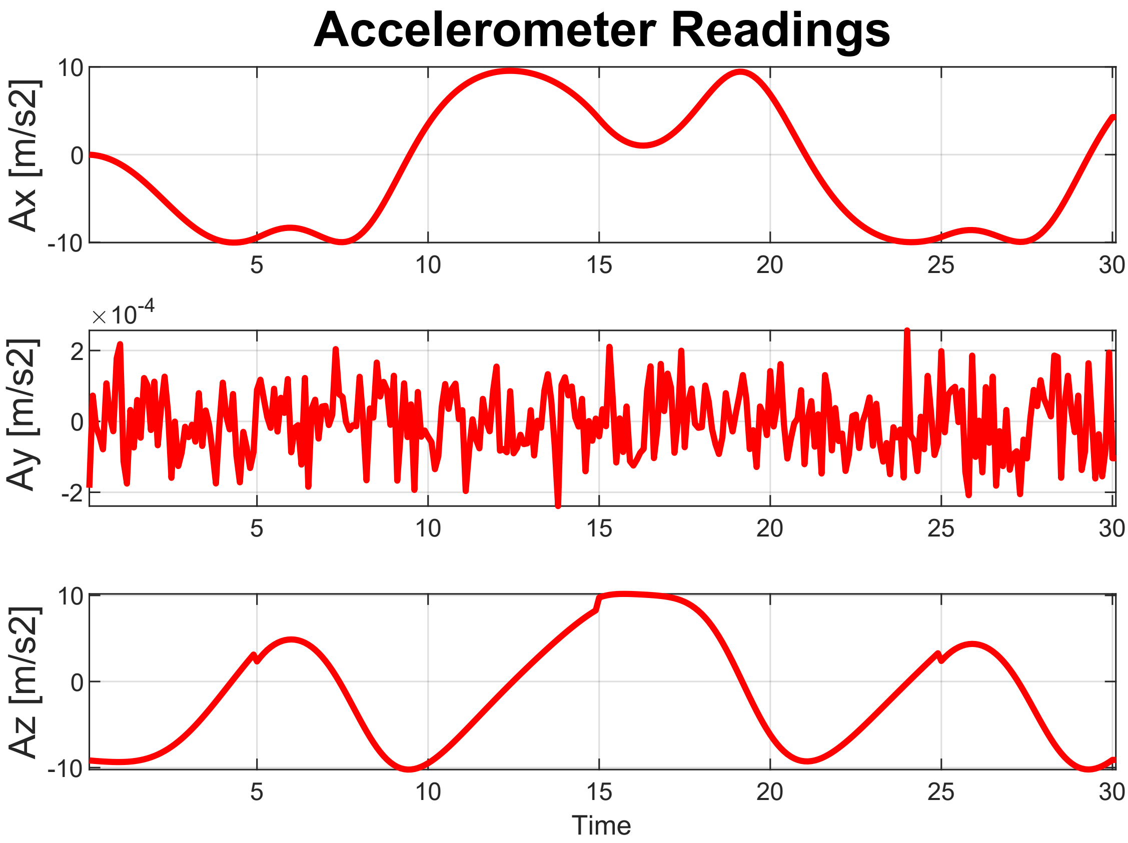


Figure 2.6: Accelerometer Measurements Simulation

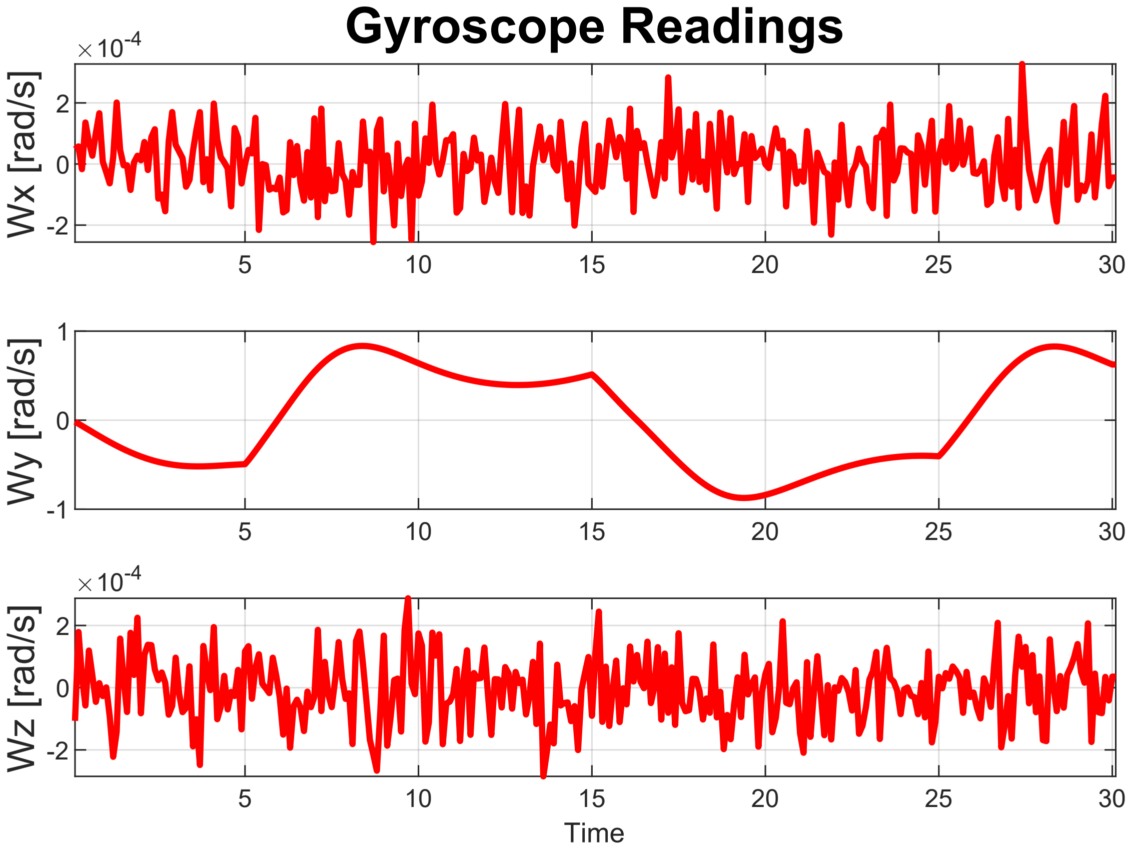


Figure 2.7: Gyroscope Measurements Simulation

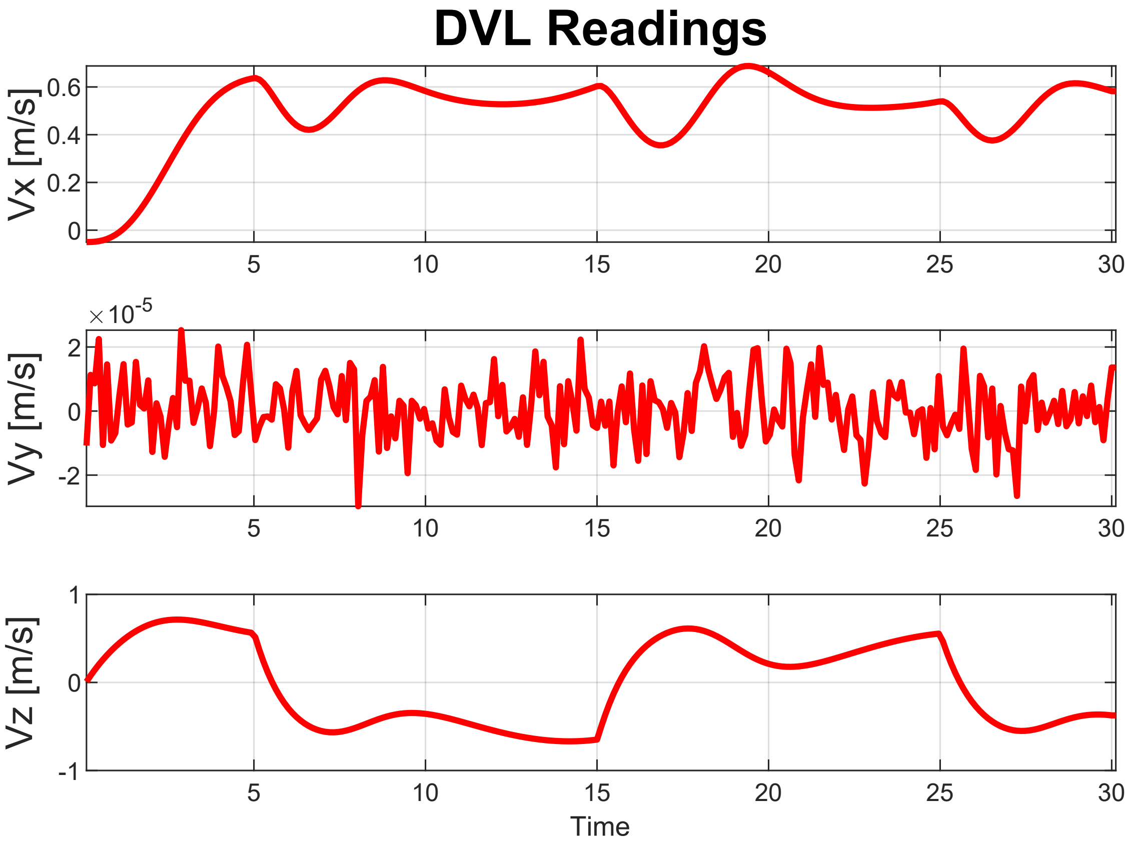


Figure 2.8: DVL Measurements Simulation

Chapter 3

Model Identification

In the previous chapter the physical model of Zeno has been presented and analyzed; it has been shown that many parameters of that model are unknown. In many cases, the parameters of a model that describes the behaviour of a system cannot be determined a priori just using some physical formula, above all when the model is describing complex situations like the fluid dynamics of a prototype. Indeed, sometimes it is impossible to find an appropriate law for the description of that specific situation. In other cases, it is impossible to fit the law to that specific case without making new hypothesis that could lead to values of the parameters too far from the real ones. In all these cases the solution is approaching to the problem with the system identification theory.

System identification is aimed at constructing and selecting mathematical models M of dynamical data generating system S. The first step is to determine a class of models M within which the search for the most suitable model is to be conducted; this can be a class of parametric models $M(\theta)$, where θ is the vector of parameters [9]. In case of TI system (Time Invariant) the values of this vector are constants. The second step is to chose the best values of this parameters. In order to do this, a parametric estimation has to be set: starting form experimental measurements of the inputs and the outputs of the system, the values of the unknown parameters can be evaluated empirically solving an optimization problem. Finally, a validation of the specific model corresponding to the estimated values has to be performed. This validation is done making a comparison between the experimental outputs and the simulated ones, feeding the real and simulated system with the same inputs. The estimated model is acceptable if real and synthetic data are "close" enough. This concept is quantified through the use of certain criteria. The meaning of optimization and some theory about it will be discussed in the first section of this chapter.

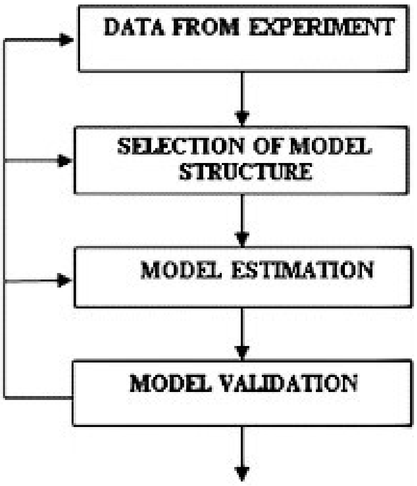


Figure 3.1: System Identification Operative Sequence

In the specific case of my work of thesis, the class models has been selected considering the physical model presented in section 1.3 and 2.2. This model is complex and coupled. For these reasons, it is difficult to set an appropriate optimization problem that could be solved. In the second section of this chapter, the theory of Linear Programming (also called Linear Optimization) and Regressors will be presented. It will be also shown how to formulate the problem of parameters estimation in this terms. In the last part of the chapter it will be shown how to reduce the problem of identifying the parameters of Zeno to a sequence of suitable linear optimization problems that can be always solved.

3.1 Optimization Problem

Optimization is a technology that can be used to devise decisions in several context. In first place, a suitable mathematical model for the description of the concrete problem has to be built; than, it has to be solved by means of suitable algorithms [10]. The mathematical model describing an optimization problem can be defined as follow:

$$\begin{aligned} \text{minimize} \quad & f_0(\boldsymbol{x}) \\ \text{subject to} \quad & f_i(\boldsymbol{x}) \leq b_i, \quad i = 1, ..., m \end{aligned} \tag{3.1}$$

Here the vector $\boldsymbol{x}$ is the optimization variable, $f_0 : \mathbb{R}^n \to \mathbb{R}$ is the cost or objective function to be minimized, in order to get the best solution, $f_i : \mathbb{R}^n \to \mathbb{R}$, are the constraint functions and b_i are the limits or bound for these constraints [11]. The feasible set of problem (3.1) is defined as follow:

$$X = \{\boldsymbol{x} \in \mathbb{R}^n s.t. : f_i(\boldsymbol{x}) \leq b_i, \quad i = 1, ..., m\} \tag{3.2}$$

The optimal solution of the problem is:

$$p^* = \min_{\boldsymbol{x} \in X} f_0(\boldsymbol{x}) \tag{3.3}$$

while the notation for the optimal set is:

$$X_{opt} = \underset{\boldsymbol{x} \in X}{\operatorname{argmin}} f_0(\boldsymbol{x}) \tag{3.4}$$

This is the most generic form for an optimization problem. This problem could have no solution. This happens, for example, when the problem is infeasible i.e if $X = \emptyset$. Other times the problem could have solutions difficult to be found. Indeed, many algorithms are meant to find local optimal, that is to say, a minimum of the objective function $f_0(\boldsymbol{x})$ only for points of the feasible set X nearby to the corresponding local optimal point $\boldsymbol{x}_{opt}$. In our case the interest is in finding the global optimal solution i.e. p^*

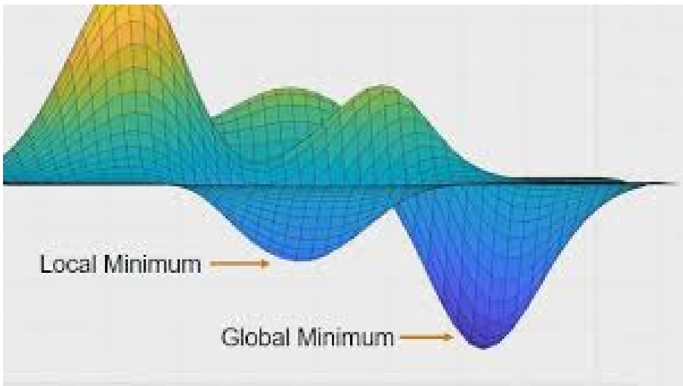


Figure 3.2: Graphical Difference Between Global and Local solution

A subgroup of the optimization problems are the convex ones (CP). A convex optimization problem is a problem in the form (3.1) where the functions $f_i(\boldsymbol{x}), \quad i = 0, ..., m$ satisfies the following property:

$$f_i(\alpha\boldsymbol{x} + \beta\boldsymbol{y}) \leq \alpha f_i(\boldsymbol{x}) + \beta f_i(\boldsymbol{y}) \tag{3.5}$$

for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ and all $\alpha, \beta \in \mathbb{R}$ with $\alpha + \beta = 1, \alpha \geq 0, \beta \geq 0$. These category of problems have a key property: any local solution is also a global solution. As a result, any solver that is able to find a local solution will find a global solution for a convex optimization problem. In figure (3.3) it is shown the typical shape of a convex optimization objective function. Looking at the figure, it is intuitive to understand why local minima coincide with global ones.

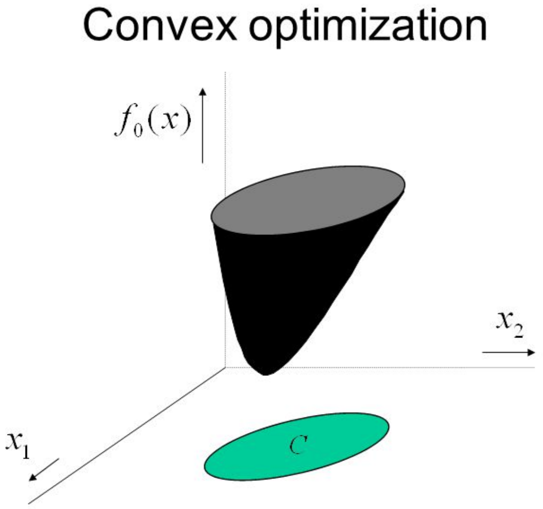


Figure 3.3: Graphic of a Convex Optimization Problem

3.2 Regressor Form

A subgroup of the CP problems are the Linear Programming problems (LP). The classical notation for LP problems is the following:

$$\begin{aligned} \text{minimize} \quad & \boldsymbol{c}^T \boldsymbol{x} \\ \text{subject to} \quad & \boldsymbol{a}_i^T \boldsymbol{x} \leq b_i, \quad i = 1, ..., m \end{aligned} \tag{3.6}$$

Here the vectors $\boldsymbol{a}_i, \boldsymbol{c} \in \mathbb{R}^n$ and scalars $b_i \in \mathbb{R}$ are constant parameters that specify objective and contraints of the problem [11]. This kind of problem can be easily solved; many solver have been developed for this aim. One example is CVX.

It is a MATLAB implemented modeling system for constructing and solving Convex optimization programs. It supports a number of standard problem types, including linear programming. CVX supports several SQLP (Semi-definite Quadratic Linear Programming) solvers, like SeDuMi and SDPT3 [12]. The MATLAB code developed for the identification of Zeno's parameters uses CVX in order to solve the sequence of optimization problems necessary for this aim.

Now it is necessary to show how to formulate a problem of parameters estimation in the form of a LP problem. Let's consider to have a polynomial data fitting in the form of

$$y = \theta_n x^n + \theta_{n-1} x^{n-1} + ... + \theta_0 \tag{3.7}$$

where y and x are measured quantities subject to noise, and $\boldsymbol{\theta} = [\theta_n, \theta_{n-1}, ..., \theta_0]$ is the vector of unknown coefficients. This problem can be seen like a problem of parameters estimation, where $\boldsymbol{\theta}$ is the vector of parameters to be estimated.
Lets consider to make a test and get from it N measurements of the quantities x and y; it is possible to obtain two vectors $\boldsymbol{X}_N = [x_1, x_2, ..., x_N]^T$ and $\boldsymbol{Y}_N = [y_1, y_2, ..., y_N]^T$. The following matrix can be built:

$$\boldsymbol{\Phi}(\boldsymbol{X}_N) = \begin{bmatrix} x_1^n & x_1^{n-1} & . & . & . & x_1 & 1 \\ x_2^n & x_2^{n-1} & . & . & . & x_2 & 1 \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ x_{N-1}^n & x_{N-1}^{n-1} & . & . & . & x_{N-1} & 1 \\ x_N^n & x_N^{n-1} & . & . & . & x_N & 1 \end{bmatrix} \tag{3.8}$$

This matrix is called *Regressor* matrix.

Considering that both measurements of x and y are subject to errors, it is possible to define

$$\boldsymbol{\epsilon}_N = \boldsymbol{\Phi}(\boldsymbol{X}_N)\hat{\boldsymbol{\theta}} - \boldsymbol{Y}_N \tag{3.9}$$

where $\hat{\boldsymbol{\theta}}$ a certain estimation of the vector of parameters. $\boldsymbol{\epsilon}_N$ is called the vector of the residual. It is evident that the value of this vector depends on the value of the specific estimation of $\boldsymbol{\theta}$: the better is the estimation of the parameters, the more the vector of residual tends to zero. So it is possible to set an optimization problem, where the objective is to minimize the residual vector. In particular, it has been chosen to minimize it in the sense of the L1 norm, so the resulting notation is:

$$\min_{\boldsymbol{\theta}} \|\boldsymbol{\Phi}(\boldsymbol{X}_N)\boldsymbol{\theta} - \boldsymbol{Y}_N\|_1 \tag{3.10}$$

This problem can be rewritten considering a vector of additional slack variables $\boldsymbol{\alpha} \in \mathbb{R}^N$. In this way the following new problem can be considered:

$$\begin{gathered} \min_{\boldsymbol{\theta},\boldsymbol{\alpha}} \sum_{i=1}^{N} \alpha_i \\ \text{subject to} \\ -\boldsymbol{\alpha} \leq \boldsymbol{\Phi}(\boldsymbol{X}_N)\boldsymbol{\theta} - \boldsymbol{Y}_N \leq \boldsymbol{\alpha} \\ \boldsymbol{\alpha} \geq \mathbf{0}^T \end{gathered} \tag{3.11}$$

Substituting the summation with the equivalent scalar product $\mathbf{1}^T\boldsymbol{\alpha}$ and making some trivial calculation in order to obtain only inequality constrains with the $\leq$ sign, it's possible to see that the problem (3.11) is a Linear Programming, with $n + N$ variables and $3N$ constraints. In the case of positive parameters, it's advisable to add an additional constraint

$$\boldsymbol{\theta} \geq \mathbf{0}^T \tag{3.12}$$

In the solution of this problem, the interest is not in the optimal value but in the optimal set θ i.e. the parameters to be estimated of the initial problem (3.7).

With the same logic it's possible to cast an estimation problem into an optimization problem considering different types of norm for the residual like the $\|.\|_2$ norm or the $\|.\|_\infty$ norm. The reason why the L1 norm has been chosen for the identification of Zeno dynamic parameters is linked to the errors in the measurements: with respect to the other norms, the L1 leads to a solution that is more robust. This means that the estimation is less sensitive to bad measurement samples. In figure (3.4) it is possible to see a visual difference between the L1 minimization and the L2 minimization; it is evident that the first one leads to a better result, being less influenced by outlier measurements. The disadvantage of using this norm is that the resulting optimization problem is computationally less efficient with respect to the other ones. Anyway, the optimization of the efficiency is not mandatory in the specific case of Zeno, having the computation of the parameters off-line [13].

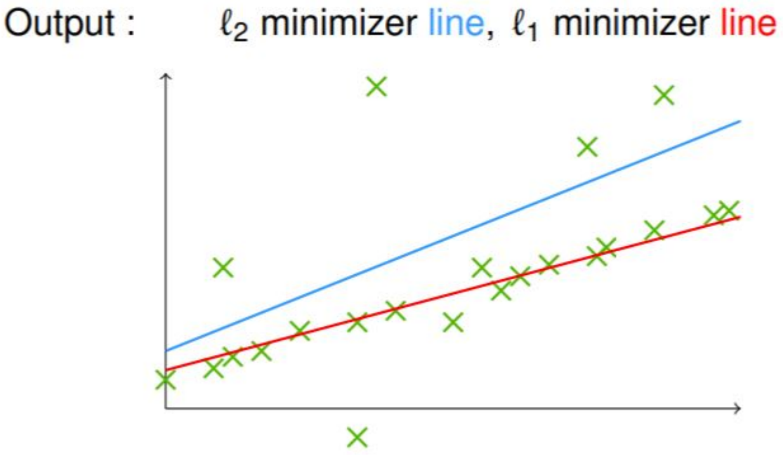


Figure 3.4: Fitting Problem Solved with L1 and L2 Norm

3.3 A Qualitative Example

Before showing how to solve the problem of the identification of Zeno parameters using the formulation just explained, a qualitative example is presented, in order to introduce a simpler dynamics parameters estimation in the form of an LP problem.

Let's consider to have an ideal vehicle of known mass m, moving only of pure surge. This motion is caused by a propulsion force directed along the x axis. Let's consider to have only linear and quadratic contribution of the dumping and to have a neutral buoyancy (gravity force equal to the buoyant force). The dynamic scalar equation describing this problem is the following:

$$\tau_u = m\dot{u} + m_{A11}\dot{u} + d_{L11}u + d_{Q11}|u|u \tag{3.13}$$

where u indicates the velocity along the surge axis. No Coriolis and centripetal terms are present, having zero angular velocity. It is considered to have two sensor of velocity and acceleration mounted on the vehicle, with the same sampling rate. During the motion, N measurements are collected, denoting them with $\dot{u}_{1,\dots,N}$ and

$u_{1,\dots,N}$. The propulsion force is known at every time instant; it is possible to built a vector $\tau_{u1,\dots,N}$ with the forces collected at the same time instant of the measurements. Even if the model is not linear with respect to the kinematics variables, in terms of parameters to be estimate the problem is linear, where the constant unknown values to be found are $X_{\dot{u}}$, d_{Q11} and d_{L11}. This estimation problem is similar to the problem (3.7); for this reason it is possible to rewrite it in the regressor form, obtaining the quantities

$$\boldsymbol{\Phi}(\boldsymbol{X}_N) = \begin{bmatrix} \dot{u_1} & u_1 & |u_1|u_1 \\ \dot{u_2} & u_2 & |u_2|u_2 \\ . & . & . \\ . & . & . \\ . & . & . \\ \dot{u_{N-1}} & u_{N-1} & |u_{N-1}|u_{N-1} \\ \dot{u_N} & u_N & |u_N|u_N \end{bmatrix} \qquad \boldsymbol{Y}_N = \begin{bmatrix} \tau_{u1} - m\dot{u}_1 \\ \tau_{u2} - m\dot{u}_2 \\ . \\ . \\ . \\ \tau_{uN-1} - m\dot{u}_{N-1} \\ \tau_{uN} - m\dot{u}_N \end{bmatrix}$$

$$\boldsymbol{\theta} = \begin{bmatrix} m_{A11} \\ d_{Q11} \\ d_{L11} \end{bmatrix} \tag{3.14}$$

The problem (3.11) can be set on the base of these matrices. Moreover, it is possible to add the constraint (3.12). Indeed all the diagonal terms of drag matrices and added mass matrices have to be positive in order to satisfy the property of being positive definite. The result will be the vector of estimation

$$\hat{\boldsymbol{\theta}} = \begin{bmatrix} \hat{m}_{A11} \\ \hat{d}_{Q11} \\ \hat{d}_{L11} \end{bmatrix} \tag{3.15}$$

3.4 Zeno AUV Parameters Identification

This section contain the description of the algorithm used for the identification of the parameters of the vehicles, in view of the discussion above.

In first place, the dynamic equations of Zeno considered in this treatment are reminded:

$$(\boldsymbol{M}_{RB} + \boldsymbol{M}_A)\dot{\boldsymbol{\nu}} + (\boldsymbol{C}_{RB}(\boldsymbol{\nu}) + \boldsymbol{C}_A(\boldsymbol{\nu}))\boldsymbol{\nu} + (\boldsymbol{D}_L + \boldsymbol{D}_Q(\boldsymbol{\nu}))\boldsymbol{\nu} + \boldsymbol{g}(\boldsymbol{\eta}) = \boldsymbol{\tau}$$

Where the explication of these matrices have been reported in section 2.2. It is clear that in general, considering the unknown parameters as variables of the equations, we are in front of a nonlinear problem high coupled. It is almost impossible to solve a non linear optimization problem with 24 coupled variables. The general idea behind my work of thesis is to try to decouple this complex problem into a certain number of simpler problems, exploiting the already existent PID controller and suitable tests.

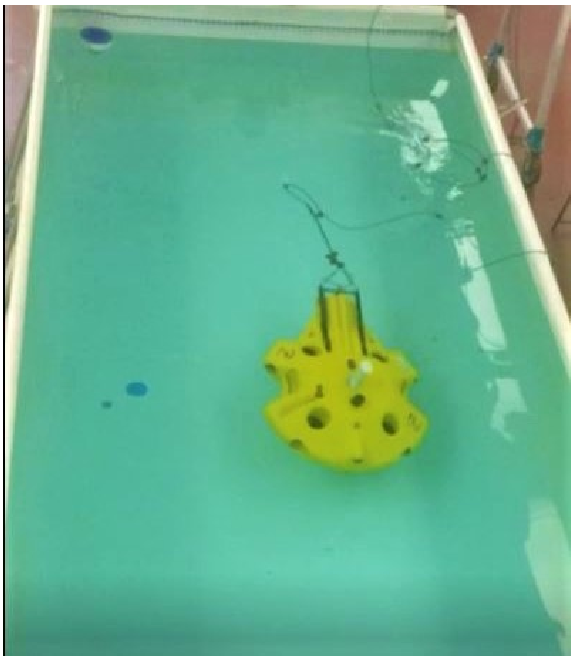


Figure 3.5: Zeno Inside the Testing Pool of the Lab

3.4.1 Linear Parameter Identification

Let's consider Zeno in a rest position and orientation (no velocity or accelerations, null values of pitch and roll angles) in a controlled environment (figure 3.5). The PID controller is activated, with the only purpose of fixing the orientation to the actual one and the heave position to a value just under the surface of the water. Then, in a certain moment a force is applied, directed only along the X axis. In first approximation, the PID controller prevents any angular motion to occur. The result is that all the coupling terms and, as a consequence, all the diagonal terms related to each motion except for the surge one, are not stimulated. From a mathematical point of view, this particular test dynamics can be described by the only first equation, in a form totally similar to the form of the equation of the example (3.13) where:

- No coupling, Coriolis or Centripetal terms are present, having only one component in the velocity and acceleration vectors.
- No components of the restoring forces vector are present, thanks to the action of the PID that forces to zero the values of RPY angles (the zero of the Yaw has been chosen arbitrarily) and balances the difference between buoyancy and gravity forces.
- The other equations can be neglected, thanks to the PID action that balanced time by time the forces caused by the coupling terms.

In the same way it is possible to plan the test along the other directions, with only few little differences. Below are reported the six simplified equations for the identification of the diagonal terms.

1. The equation for the dynamic during the Surge test is

$$\tau_u = m\dot{u} + m_{A11}\dot{u} + d_{L11}u + d_{Q11}|u|u \tag{3.16}$$

   As described above.

2. The equation for the dynamic during the Sway test is

$$\tau_v = m\dot{v} + m_{A22}\dot{v} + d_{L22}v + d_{Q22}|v|v \tag{3.17}$$

   No differences occur with respect to the previous equation.

3. The equation for the dynamic during the Heave test is

$$\tau_w = m\dot{w} + m_{A33}\dot{w} + d_{L33}w + d_{Q33}|w|w + (B - W) \tag{3.18}$$

where B and W are respectively the third component of the buoyancy force and of the gravity force. In this case, in fact, the action of the PID that fixed the heave position has to be switched off. As a result, the restoring contribute has to be considered.

4. The equation for the dynamic during the Roll test is

$$\tau_p = I_x\dot{p} + m_{A44}\dot{p} + d_{L44}p + d_{Q44}|p|p - z_B B sen(\phi) \tag{3.19}$$

where z_B is the third component of the position of the buoyancy center $\boldsymbol{r}_B$ and ϕ is the roll angle. In all the three rotation tests, in fact, the action of the PID fixes the position of Zeno, letting the three angle free to move. The results are the same for what concern the coupling contribution, the only difference is that in this case the torque produced by the restoring action has to be considered.

5. The equation for the dynamic during the Pitch test is

$$\tau_q = I_y\dot{q} + m_{A55}\dot{q} + d_{L55}q + d_{Q55}|q|q - z_B B sen(\theta) \tag{3.20}$$

where θ is the pitch angle.

6. The equation for the dynamic during the Yaw test is

$$\tau_r = I_z\dot{r} + m_{A66}\dot{r} + d_{L66}r + d_{Q66}|r|r \tag{3.21}$$

No restoring contribution is present because any change in the yaw orientation doesn't affect the parallelism between $\boldsymbol{r}_B$ and $\boldsymbol{f}_B$.

All this equations are linear with respect to the unknown parameters so 6 optimizations problem in the form (3.11) can be set and solved independently one from the other. The resulting estimated $\boldsymbol{\theta}_i$ can be reordered so that it is possible to obtain the following three estimated matrices:

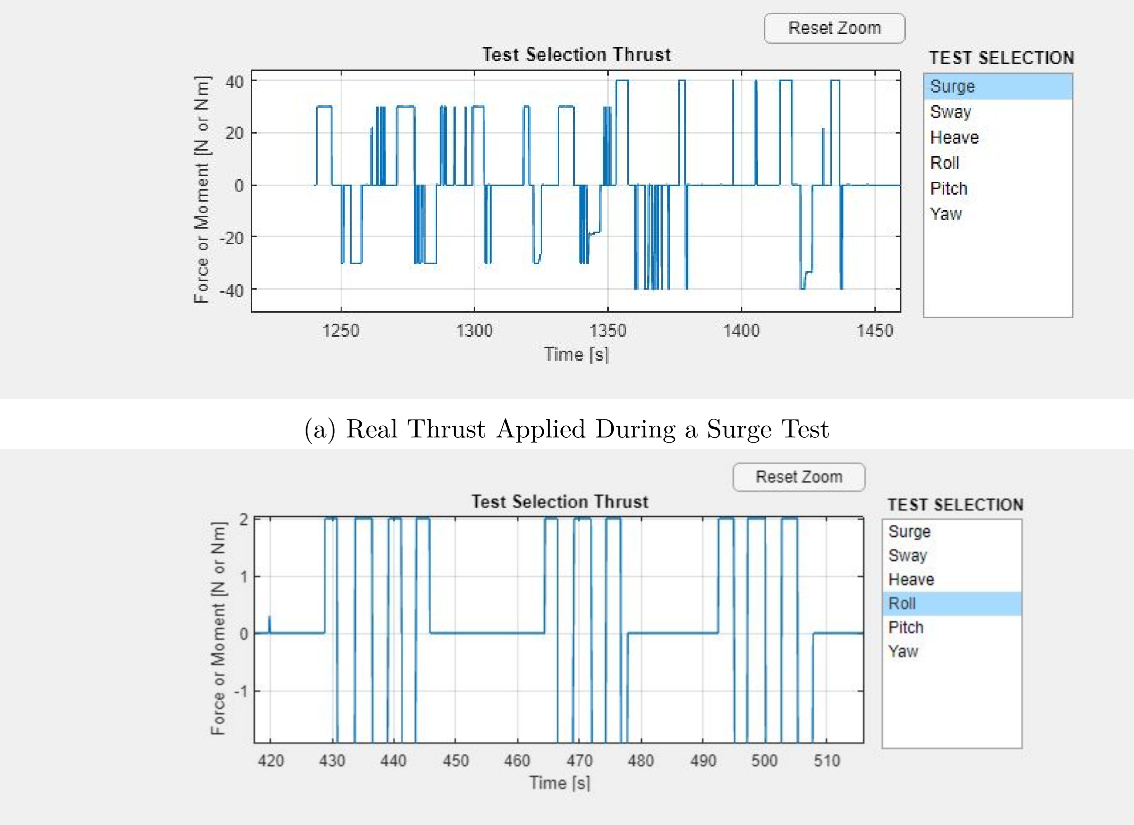
$$\hat{\boldsymbol{M}}_A = \begin{bmatrix} \hat{m}_{A11} & 0 & 0 & 0 & 0 & 0 \\ 0 & \hat{m}_{A22} & 0 & 0 & 0 & 0 \\ 0 & 0 & \hat{m}_{A33} & 0 & 0 & 0 \\ 0 & 0 & 0 & \hat{m}_{A44} & 0 & 0 \\ 0 & 0 & 0 & 0 & \hat{m}_{A55} & 0 \\ 0 & 0 & 0 & 0 & 0 & \hat{m}_{A66} \end{bmatrix} \tag{3.22}$$

$$\hat{\boldsymbol{D}}_L = \begin{bmatrix} \hat{d}_{L11} & 0 & 0 & 0 & 0 & 0 \\ 0 & \hat{d}_{L22} & 0 & 0 & 0 & 0 \\ 0 & 0 & \hat{d}_{L33} & 0 & 0 & 0 \\ 0 & 0 & 0 & \hat{d}_{L44} & 0 & 0 \\ 0 & 0 & 0 & 0 & \hat{d}_{L55} & 0 \\ 0 & 0 & 0 & 0 & 0 & \hat{d}_{L66} \end{bmatrix} \tag{3.23}$$

$$\hat{\boldsymbol{D}}_Q = \begin{bmatrix} \hat{d}_{Q11} & 0 & 0 & 0 & 0 & 0 \\ 0 & \hat{d}_{Q22} & 0 & 0 & 0 & 0 \\ 0 & 0 & \hat{d}_{Q33} & 0 & 0 & 0 \\ 0 & 0 & 0 & \hat{d}_{Q44} & 0 & 0 \\ 0 & 0 & 0 & 0 & \hat{d}_{Q55} & 0 \\ 0 & 0 & 0 & 0 & 0 & \hat{d}_{Q66} \end{bmatrix} \tag{3.24}$$

It is important to state that the action of the PID controller has two unwanted consequences. The first is that the controller is not able in general to maintain the reference values. For this reason, some oscillation could occur during the test. Anyway, this error is small, because the time reaction of the PID is way more faster than the mechanical time reaction of the vehicle. For this reason these oscillations have been considered as if they were measuring errors. The second consequence is that, in order to reject these oscillations, the PID could generate an additional propulsion in the direction that we are testing. Anyway, it has been observed that this contribution is in general really small if the vehicle isn't subjected to high values of propulsion, and so can be neglected.

Taking into account the previous consideration and having that, in order to excite all the dynamic behaviours of the vehicle, it is better to apply step variations of the input, the shape of the thrusts applied during each test has been chosen to be rectangular (3.6), with variable time duration and limited amplitude (50% of the maximum value that the motors could applied).



(a) Real Thrust Applied During a Surge Test

(b) Real Thrust Applied During a Roll Test

Figure 3.6: Some Examples of Real Propulsion

3.4.2 Coupling Parameter Identification

Once the first phase of the identification algorithm have been applied, 18 of the 24 unknown terms have been estimated. Assuming that the estimated values are close to the real ones, the second phase of the algorithm is based on the usage of these values as known terms for a new group of optimization problems. Our goal is to identify the coupling terms of Heave-Pitch and Sway-Yaw.

The procedure is the same for both the cases: a first optimization problem is set, having only one unknown parameter, the off-diagonal mass term (remembering that the added mass matrix has been assumed symmetric). This situation is likely, considering, as input, a rectangular force along the Heave direction with a high frequency. In this way, the vehicle will keep a low value of velocity and an high value of acceleration. As a consequence, the drag influence in the dynamics of this test will be negligible. The PID controller allows the vehicle to move along the Z

axis and to rotate along the Y axis during the Heave-Pitch test; along the Y axis and to rotate along the Z axis during the Sway-Yaw test. All the other 4 DoF are fixed to a certain rest value. In this way is possible to consider only the two equation related to the two free directions.
Than, a second optimization problem is set, considering the just found value of mass as a new known parameter. This new problem will have as variables, two off-diagonal drag terms, one quadratic and one linear (remembering that the drag matrices have been assumed symmetric). A new test in similar conditions has to be performed. This time the rectangular thrust will have a lower frequency, in order to excite highly the drag contribution.

1. The equations for the dynamic of the Pitch-Heave test are

$$\begin{aligned} m_{A53}\dot{w} &= \tau_w - (m_{A33} + m)\dot{w} - (d_{L33} + d_{Q33}|w|)w + \\ &- (B - W)cos(\theta) \\ \\ m_{A53}\dot{q} &= \tau_q - (m_{A55} + I_y)\dot{q} - (d_{L55} + d_{Q55}|q|)q + \\ &+ z_B B sin(\theta) \end{aligned} \tag{3.25}$$

   for the identification of $m_{A53} = m_{A35}$ and

$$\begin{aligned} d_{L53}p + d_{Q53}|w|p &= \tau_w - m_{A53}\dot{w} - (m_{A33} + m)\dot{w} + \\ &- (d_{L33} + d_{Q33}|w|)w - (B - W)cos(\theta) \\ \\ d_{L53}w + d_{Q53}|p|w &= \tau_q - m_{A53}\dot{q} - (m_{A55} + I_y)\dot{q} + \\ &- (d_{L55} + d_{Q55}|q|)q + z_B B sin(\theta) \end{aligned} \tag{3.26}$$

   for the identification of $d_{L53} = d_{L35}$ and $d_{L53} = d_{L35}$.

2. The equations for the dynamic of the Yaw-Sway test are

$$\begin{aligned} m_{A62}\dot{v} &= \tau_v - (m_{A22} + m)\dot{v} - (d_{L22} + d_{Q22}|v|)v \\ \\ m_{A62}\dot{r} &= \tau_r - (m_{A66} + I_z)\dot{r} - (d_{L66} + d_{Q66}|r|)r \end{aligned} \tag{3.27}$$

for the identification of $m_{A62} = m_{A26}$ and

$$\begin{aligned} & d_{L62} r + d_{Q62} |v| r = \tau_v - m_{A62} \dot{r} - (m_{A22} + m) \dot{v} + \\ & - (d_{L22} + d_{Q22} |v|) v \\ & \\ & d_{L62} v + d_{Q62} |r| v = \tau_r - m_{A62} \dot{v} - (m_{A66} + I_z) \dot{r} + \\ & - (d_{L66} + d_{Q66} |y|) y \end{aligned} \tag{3.28}$$

for the identification of $d_{L62} = d_{L26}$ and $d_{L62} = d_{L26}$.

Before performing the tests on the real vehicle, a good number of simulation has been run, with good results. Below are reported some graphical results related to the simulation of a test for the identification of the Heave-Pitch parameters, where the values of the parameters have been chosen reasonably. It is possible to observe that the validation trajectory and the simulated one are nearly superimposed, and the parameters are identified with a good accuracy (as an example, choosing $m_{A53} = 3$ the algorithm returns $\hat{m}_{A53} = 3.07$).

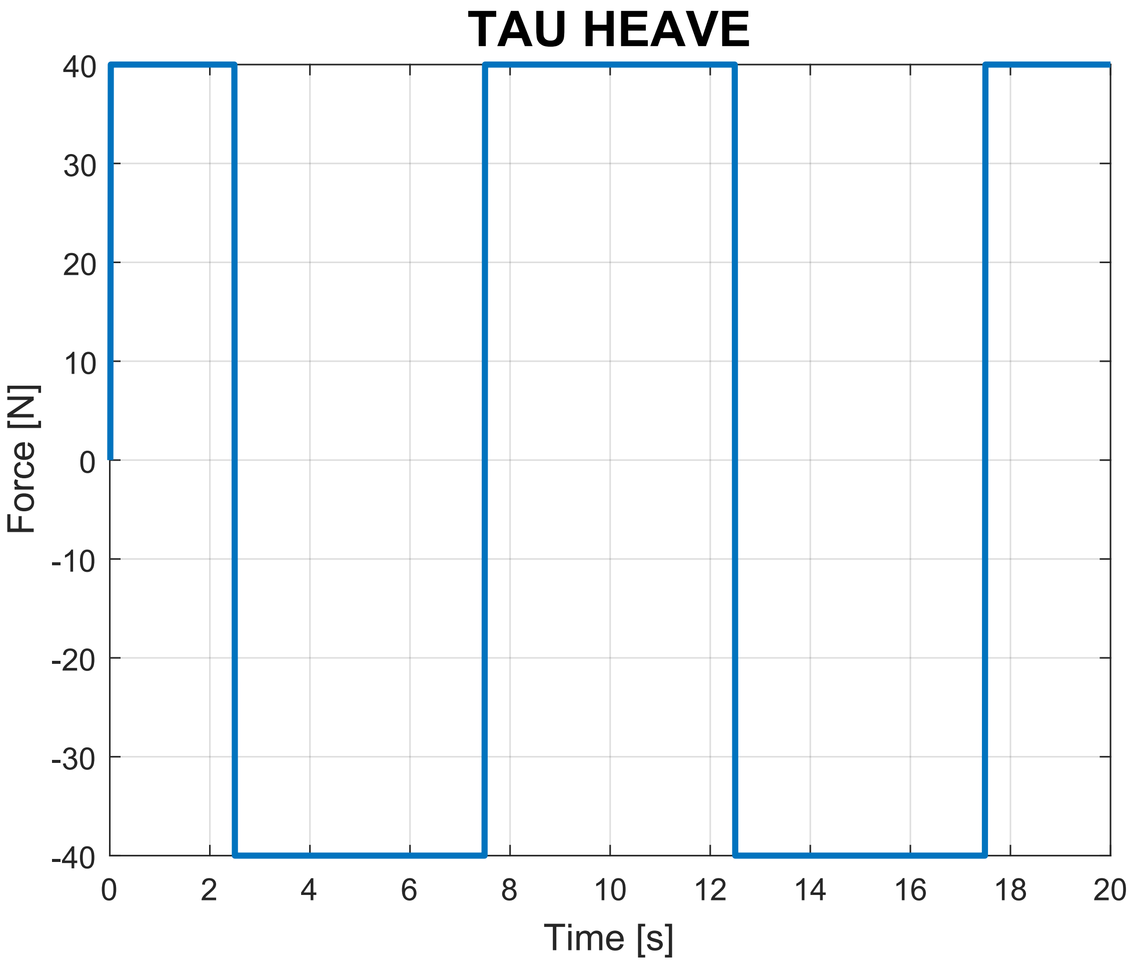


Figure 3.7: Simulated Thrust Applied Along the Z Axis

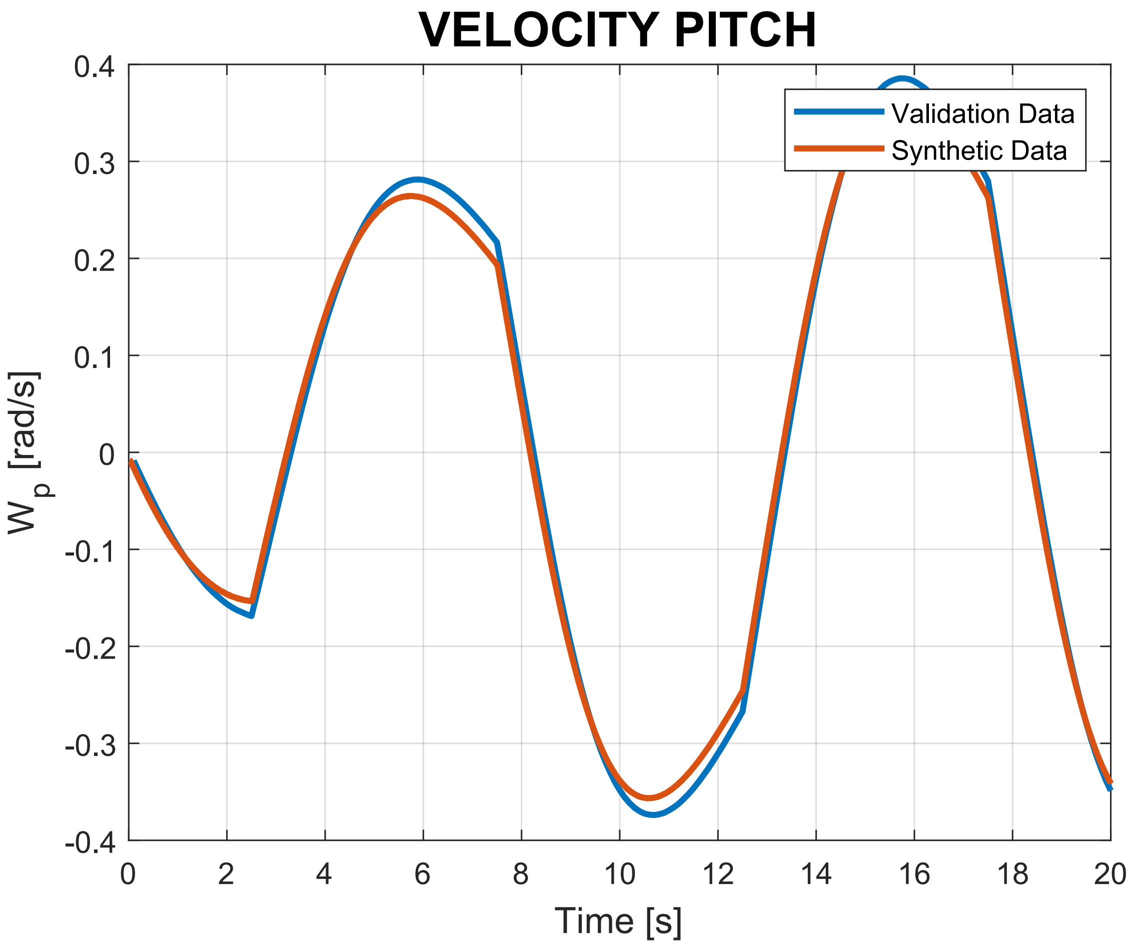


Figure 3.8: Simulation of Synthetic Pitch Velocity VS Validation Pitch Velocity

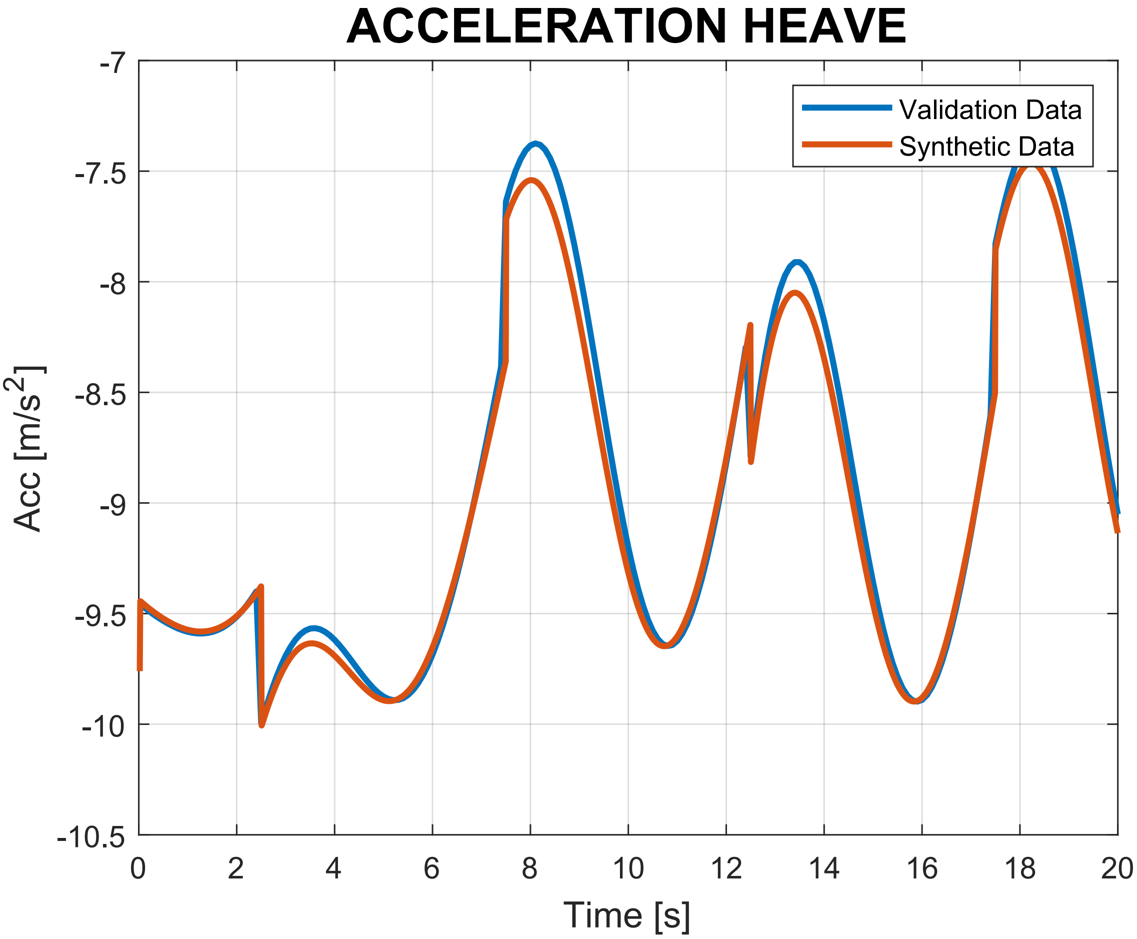


Figure 3.9: Simulation of Synthetic Heave Acceleration VS Validation Heave Acceleration

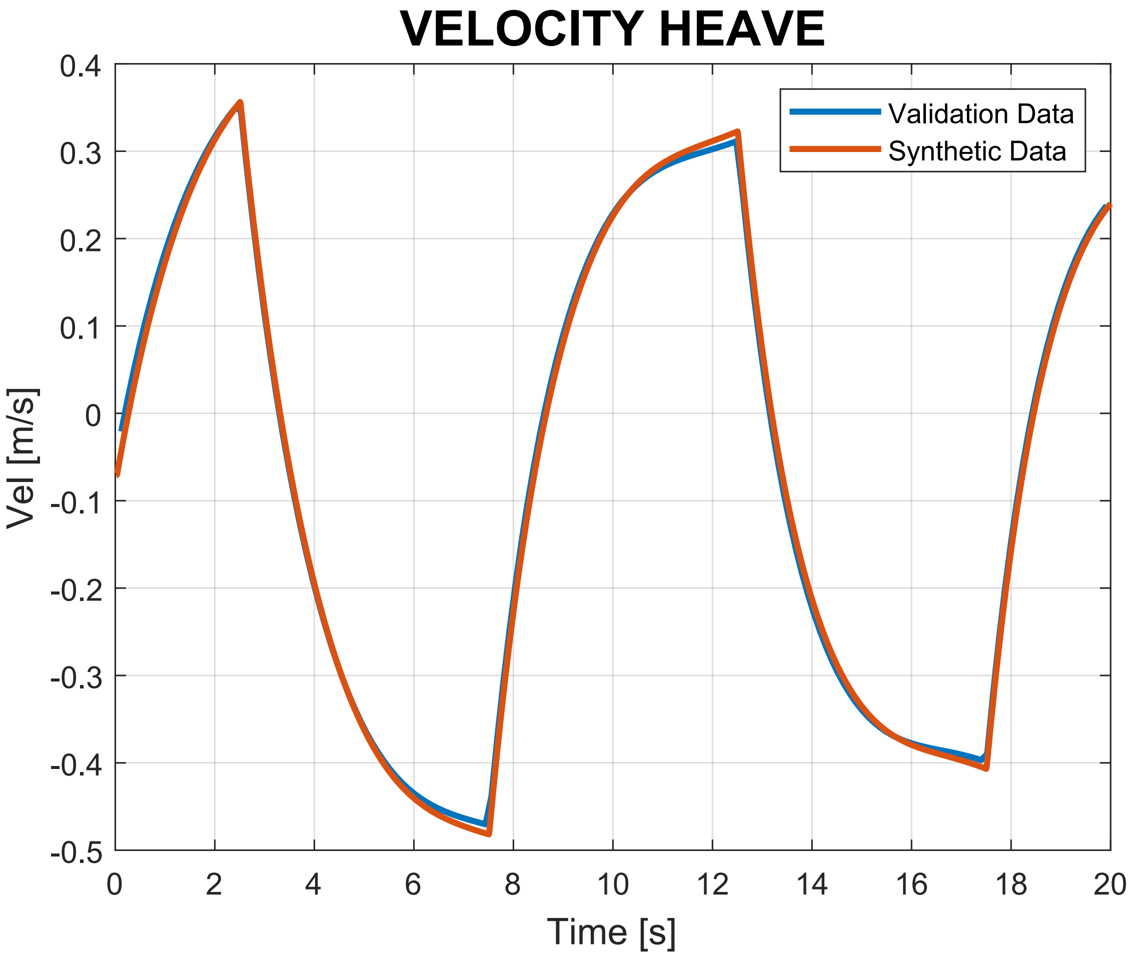


Figure 3.10: Simulation of Synthetic Heave Velocity VS Validation Heave Velocity

Chapter 4

Results

This chapter deals with the results obtained running the identification algorithms, implemented with MATLAB. The data sets collected during the test performed with the real vehicle have been used.

Some considerations have to be done. In first place, the data sets contain the measurements of the IMU and of the DVL. These data have to be pre-processed before they could be used inside the algorithms. In particular:

- The acceleration measurements contain the contribute of the gravity term. This term has to be corrected in order to get the acceleration in the fixed-reference frame, using the equation (2.3).
- The two sensors work with different sampling rates (10 Hz for the IMU, 8 Hz for the DVL). The measurements must be synchronized at the lower frequency, using an interpolation method.
- A low pass filter have to be applied in order to reduce the noise. A zero phase filtering procedure can be used so that no time delay is introduced.

Unfortunately, even applying these corrections, the measurements of acceleration are too imprecise to be used in the identification process. The reason is linked to the angular estimator. Indeed, the maximum linear acceleration of Zeno is about 1 m/s^2 so, in most of the cases, the gravity acceleration is way more bigger than the actual acceleration of the vehicle. In this condition, an error in the angular position estimation of only 0.01 rad leads to an error in the acceleration correction of, at least, 10%. In order to be able to use the measures of the accelerometers, the angular estimation should be improved.

As a consequence, it is necessary to get an estimation of linear and angular acceleration, starting from the measurements of linear and angular velocity. In this case, a simple numerical differentiation leads to quite good results, thanks to the high precision of both DVL and FOG sensors.

The last consideration is about the identification of the coupling terms. The algorithm designed for this purpose depends on the previous estimation. Even a small error in the estimation of the diagonal terms could result into a big error in the estimation of the coupling ones. The goodness of the estimation than, depends on the goodness of the measurements and, as stated before, at the moment, the acceleration is obtained with numerical differentiation, with a consequent numerical approximation error. As a result, in the next sections it's shown that, in the case of diagonal terms, good results have been achieved while, regarding the off-diagonal terms identification, measurement acquisition (linear acceleration) needs to be improved before good results can be obtained.

From the overall data set obtain from the tests, a part of the measurements is used for the identification, an other part is used for the validation. Different selections of the identification data set have been tried in order to maximize the goodness of the result.
Different criteria can be used in order to validate the result. In this treatment three of them are present:

- A qualitative validation through a plot that shows the comparison between a validation data set (Linear velocities in case of Surge, Sway, Heave tests, Angular velocities in case of Roll, Pitch, Yaw tests) and a synthetic data set. This is generated with a simulation, subject to the same input thrust of the validation test.
- A numerical criterion called "Best Fit Index" [14]. It is defined as

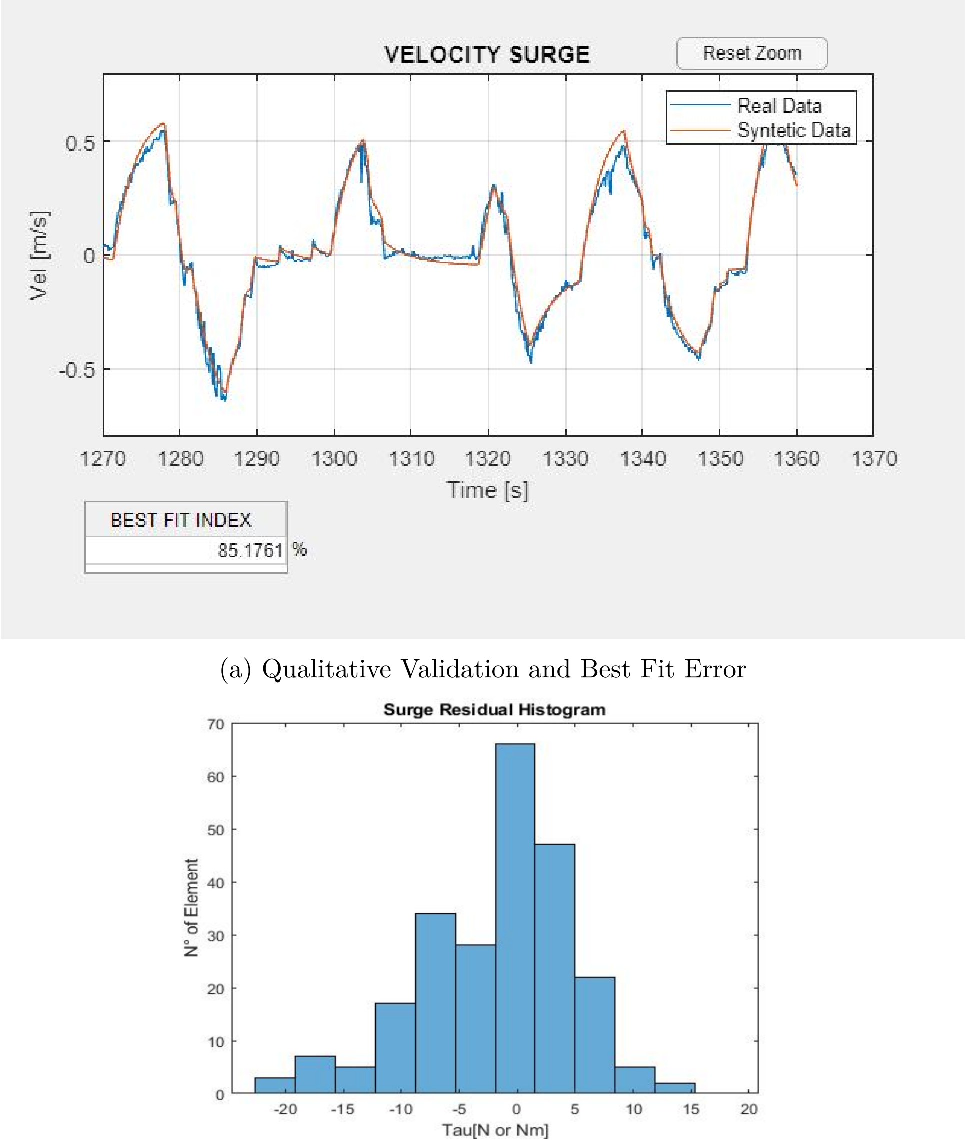
$$\boldsymbol{BFI} = 1 - \frac{\boldsymbol{RMSE}_Y}{\boldsymbol{\sigma}_Y} \tag{4.1}$$

where $\boldsymbol{RMSE_Y} = \sqrt{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}$ is the Root Mean Square Error between the validation measurements $\boldsymbol{y}_N$ and the synthetic ones $\hat{\boldsymbol{y}}_N$, while $\boldsymbol{\sigma}_Y =$

$\sqrt{\sum_{i=1}^{N}(y_i - \overline{y})^2}$ is the standard deviation of the measurements. In percentage, this value tends to 100 % when the difference between real and synthetic data tends to zero. This happen when the estimated parameters are equal to the real ones.

- An Histogram of the residual defined in equation (3.9) is presented. If its shape differs too much from a Gaussian-like form with null mean value, it could be indicative of unmodeled dynamics.

4.1 Surge



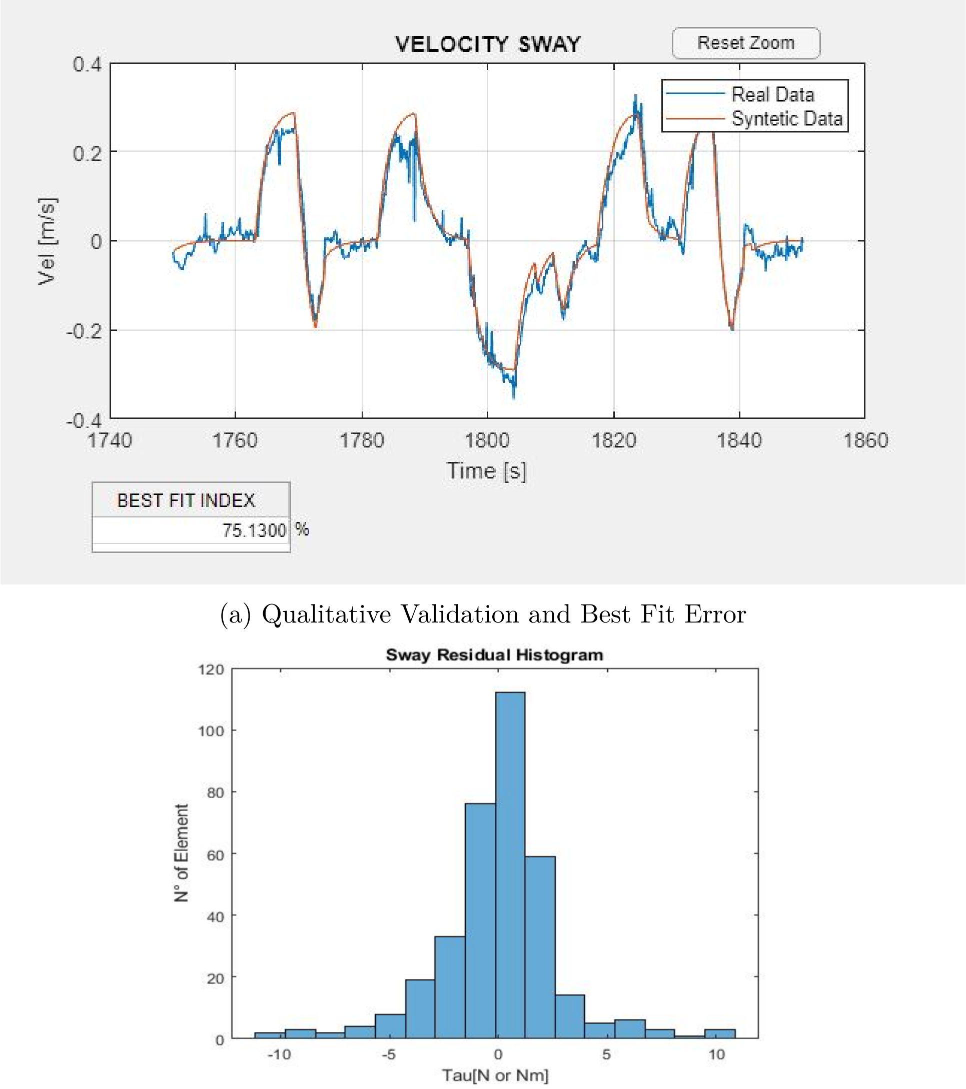
BEST FIT INDEX
85.1761 %

(a) Qualitative Validation and Best Fit Error

(b) Histogram of the Residual

Figure 4.1: Surge Validation

4.2 Sway



BEST FIT INDEX

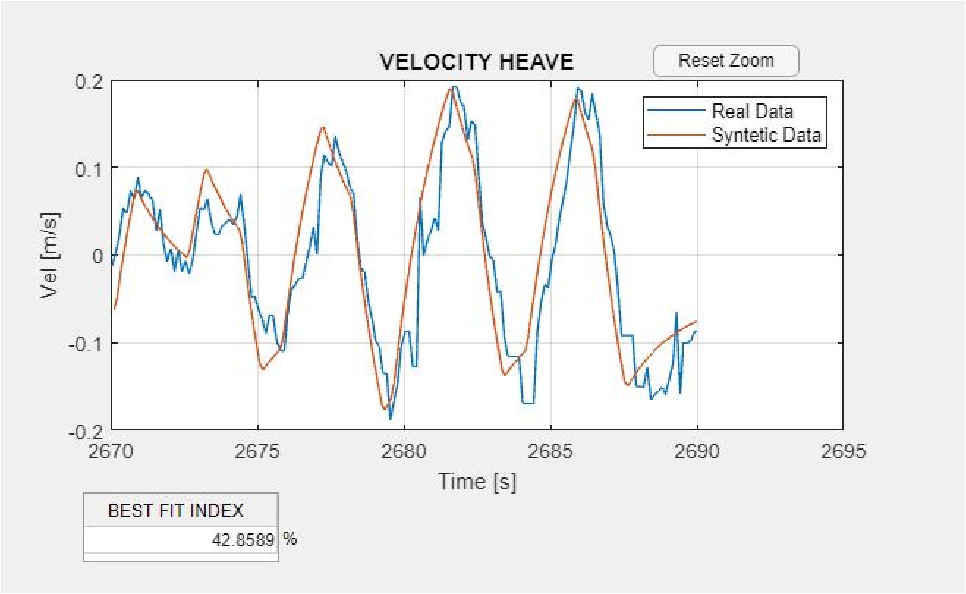
75.1300 %

(a) Qualitative Validation and Best Fit Error

(b) Histogram of the Residual

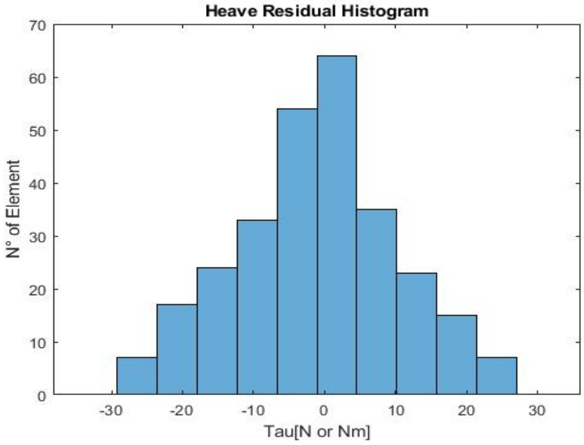
Figure 4.2: Sway Validation

4.3 Heave



BEST FIT INDEX 42.8589 %

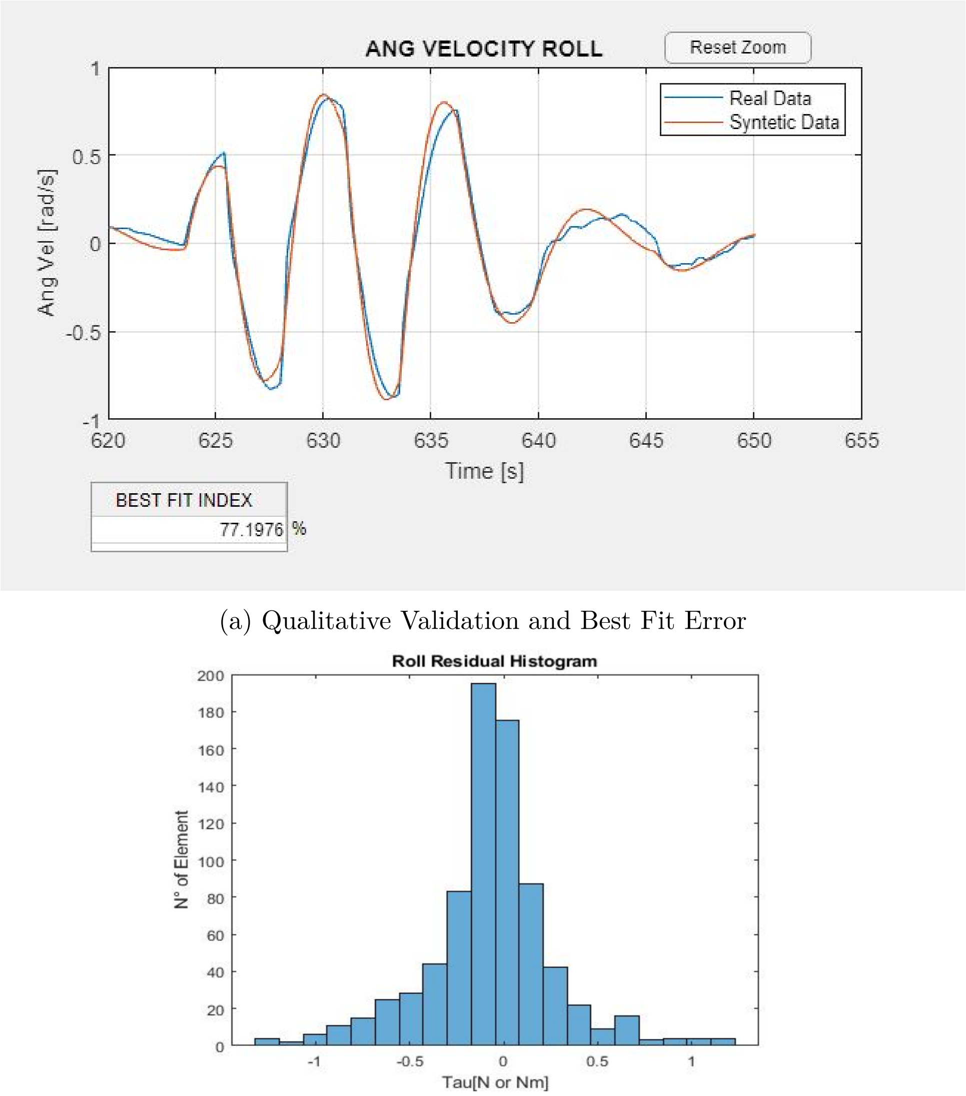
(a) Qualitative Validation and Best Fit Error



(b) Histogram of the Residual

Figure 4.3: Heave Validation

4.4 Roll



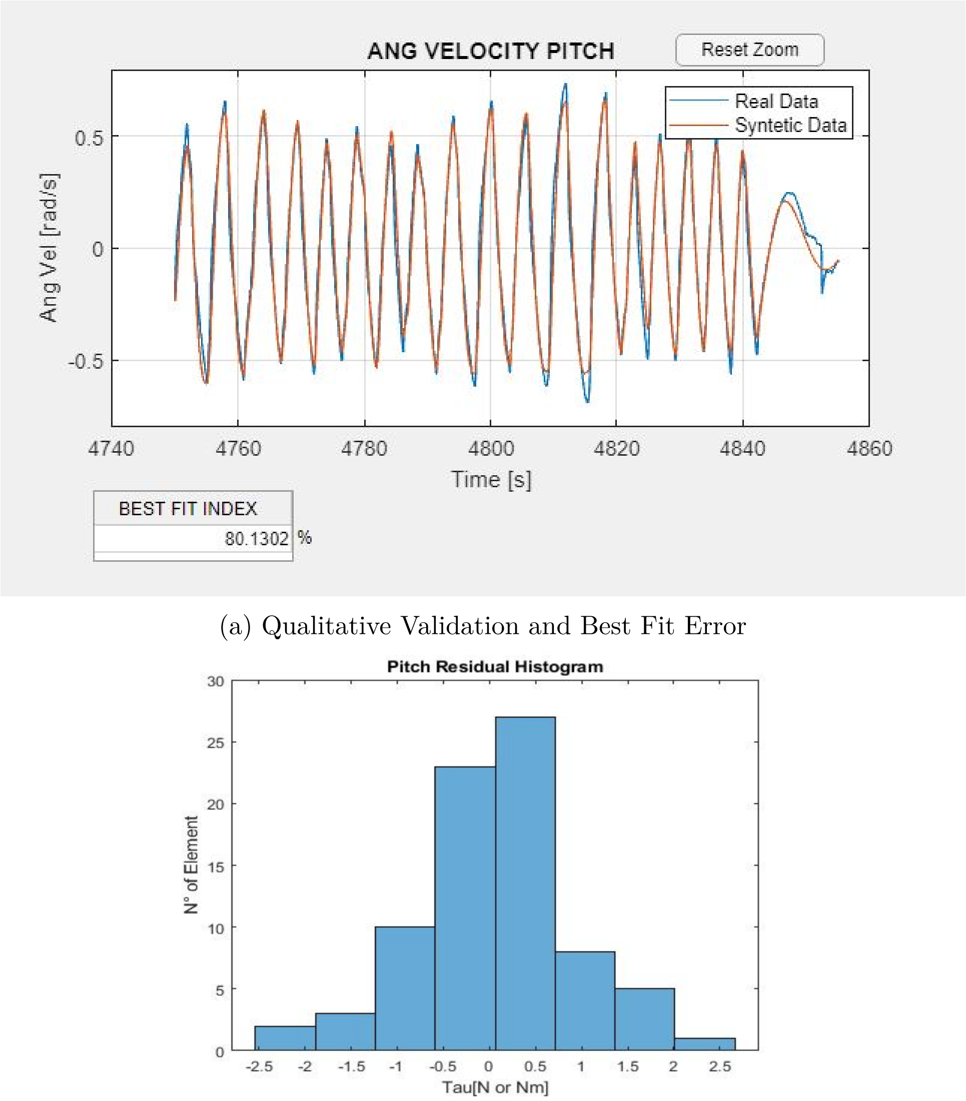
BEST FIT INDEX
77.1976 %

(a) Qualitative Validation and Best Fit Error

(b) Histogram of the Residual

Figure 4.4: Roll Validation

4.5 Pitch



BEST FIT INDEX

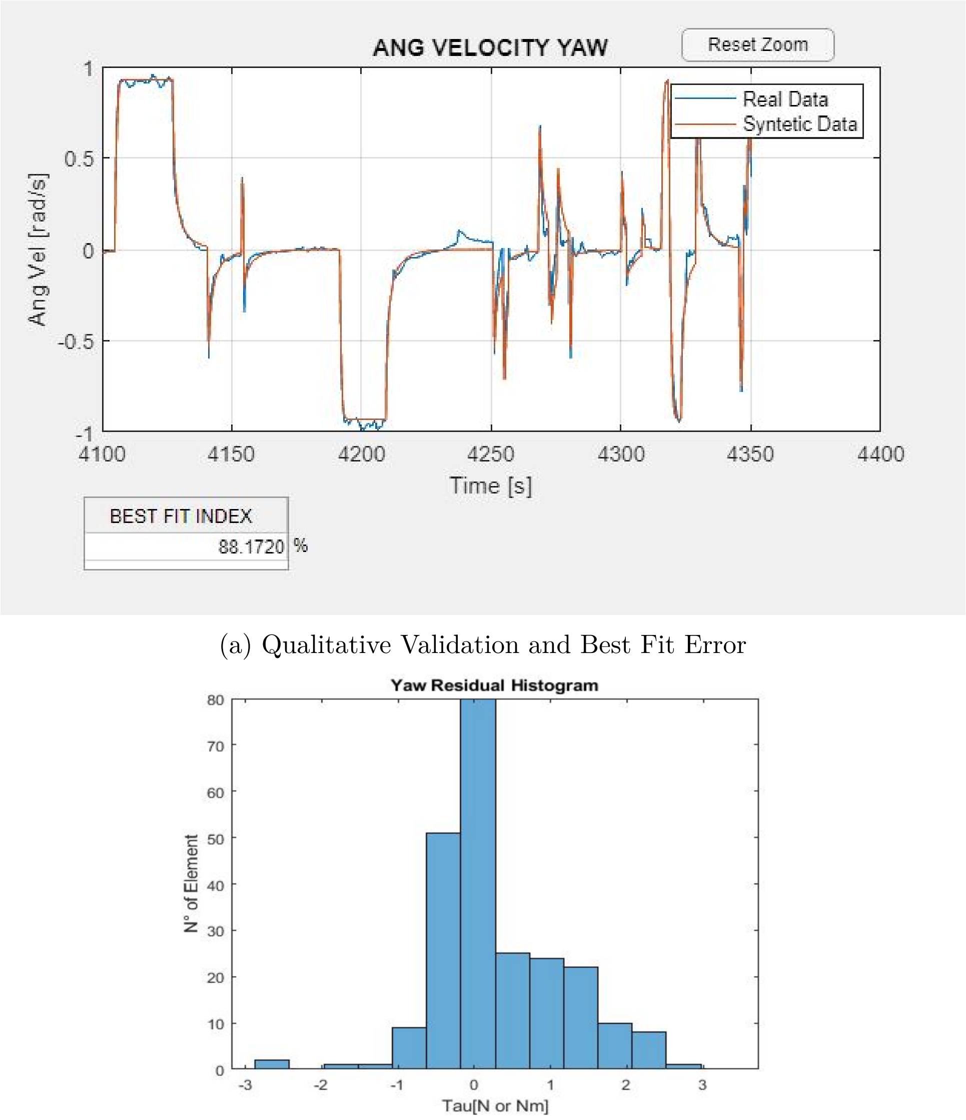
80.1302 %

(a) Qualitative Validation and Best Fit Error

(b) Histogram of the Residual

Figure 4.5: Pitch Validation

4.6 Yaw



BEST FIT INDEX

88.1720 %

(a) Qualitative Validation and Best Fit Error

(b) Histogram of the Residual

Figure 4.6: Yaw Validation

4.7 Final Result

The validation process (figures 4.1, 4.2, 4.3, 4.4, 4.5, 4.6) confirms good results in the identification of the parameters. Moreover, a good choice of the model is confirmed by the residual histograms (In first approximation, a shape similar to a Gaussian with zero mean can be deduced). The only exception appears regarding the heave test (BFI $\approx 50\%$). The reason is connected to the fact that during the test, the vehicle needs to reach positions too close to the bottom of the pool. In this condition, the DVL is not always able to take a measurement, for reasons linked to his working principle (Doppler effect). A new test should be performed in a different controlled environment with a deeper bottom, in order to get better velocity measurements. The identified values are shown below, within their respective matrices.

$$\hat{\boldsymbol{M}}_A = \begin{bmatrix} 87.75 & 0 & 0 & 0 & 0 & 0 \\ 0 & 72.63 & 0 & 0 & 0 & 0 \\ 0 & 0 & 85.07 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2.30 & 0 & 0 \\ 0 & 0 & 0 & 0 & 5.16 & 0 \\ 0 & 0 & 0 & 0 & 0 & 5.08 \end{bmatrix} kg \tag{4.2}$$

$$\hat{\boldsymbol{D}}_L = \begin{bmatrix} 30.31 & 0 & 0 & 0 & 0 & 0 \\ 0 & 59.68 & 0 & 0 & 0 & 0 \\ 0 & 0 & 72.60 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.84 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.29 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.32 \end{bmatrix} \frac{kg}{s} \tag{4.3}$$

$$\hat{\boldsymbol{D}}_Q = \begin{bmatrix} 21.09 & 0 & 0 & 0 & 0 & 0 \\ 0 & 29.52 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.59 & 0 & 0 \\ 0 & 0 & 0 & 0 & 7.49 & 0 \\ 0 & 0 & 0 & 0 & 0 & 10.14 \end{bmatrix} \frac{kg}{s} \tag{4.4}$$

Chapter 5

Applications

5.1 Control Based Design

The model identified thanks to the algorithm described in the previous chapter, can be used in several application. It is possible to use it for the simulation of vehicle dynamics. It is also possible to use for the design of a Model-Based controller, as described in section 1.2. A summary description of the controller used on Zeno will be reported. In figure (5.1), it is shown a block scheme of it.

First of all, it's possible to notice that a reference generator is necessary, in order to get the desired position, velocity and acceleration at each time instant. There are several way to implement it; a possibility is to use a third order filter. It is saturated, considering the maximum values of velocities and accelerations that Zeno can reach.

The reference signals, i.e. the position, velocity and acceleration desired, are now used as inputs for the controller. The position and the velocity are compared with the measured velocity and position (available thanks to odometry) in order to generate the PID terms. The velocity and the acceleration are used as inputs for the estimated added mass and the estimated drag terms, in order to generate the feedforward contribution. Finally, the gravity, Coriolis and centripetal terms are calculated using the measured velocity and orientation. As a result, they are going to cancel those terms in the dynamics of the Vehicle.

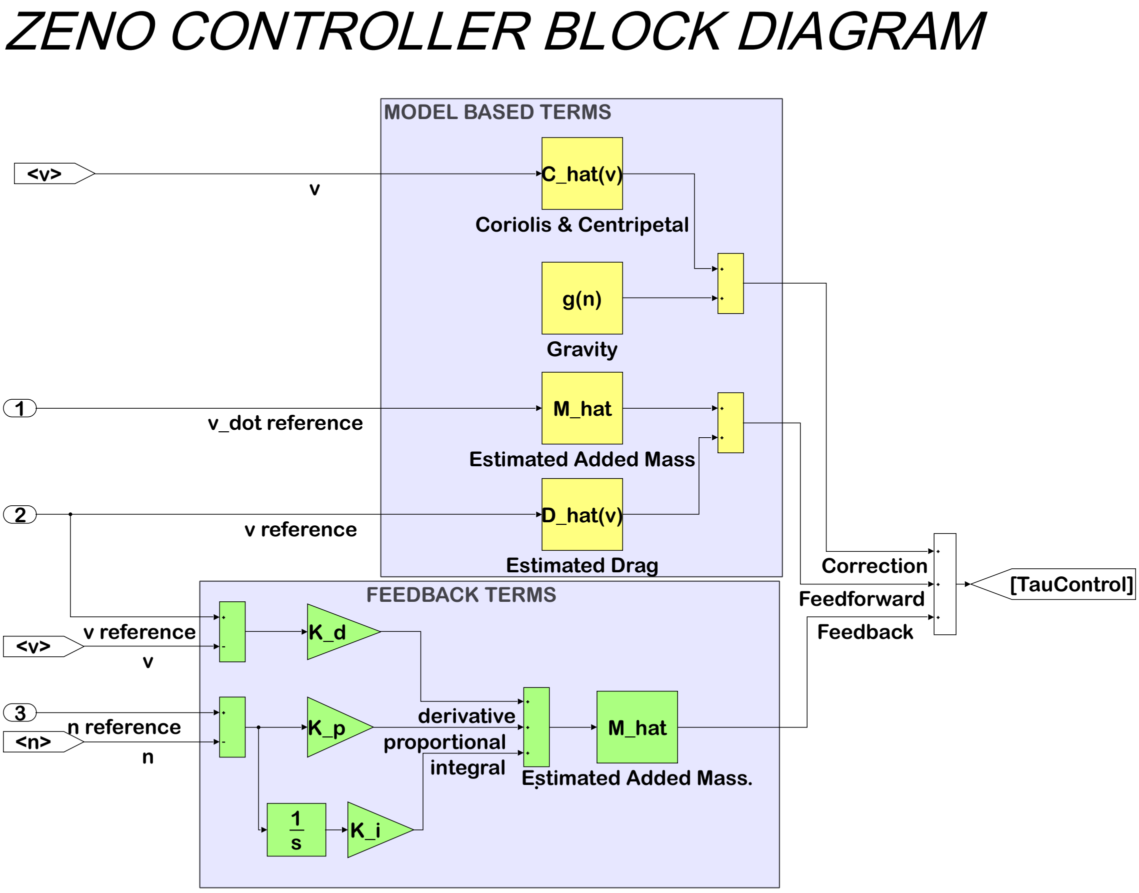


Figure 5.1: Block Scheme of the Structure of the Controller Implemented on Zeno

5.2 Results

Several test has been performed with the aim of checking the effective improvement of the control action. The Team is very satisfied by the results. In figure (5.2), one of this test is presented, showing the reference tracking performances with respect to a yaw rotation with high acceleration and speed.

In the first part of the plot, the control was performed only by the PID terms; it's possible to observe a delay in the transitory and a considerable. As a result, errors up to 20 degrees in the tracking error are reached. In the second part of the plot, the feedforward action have been activated. It's easy to observe a marked improvement in the performances. The delay in the transitory is negligible and the tracking error never exceeds 10 degrees.

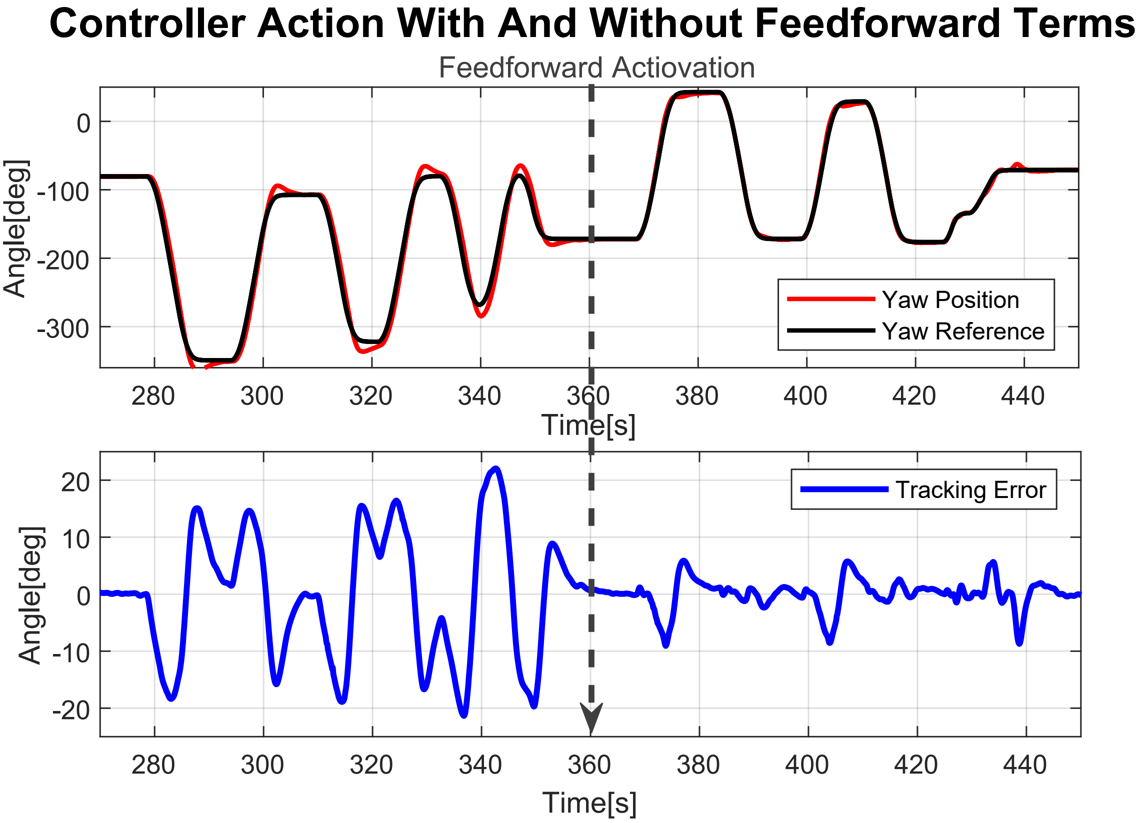


Figure 5.2: Improvement of the Control Action Using A Model Based Controller

Conclusions

The thesis work focused on studying and identifying the parameters of the dynamic model of Zeno, a vehicle developed by the "MDM Team".

This work of thesis led to useful results for the company. Mainly, the dynamic model of Zeno has been identified well enough to be used inside a new model-based control system, which ensures better performances of the vehicle. Moreover, the algorithm has been implemented in order to be as generic as possible. For this reason, it is reasonable to think about future applications for the identification of the model for new prototypes, also thanks to a Graphic Interface, easy to be used even with few notions regarding the identification of systems. More information about this interface are present in appendix A.

Several things can also be improved. The possibility of using acceleration measurements could allow significant improvements: in fact, it may be possible to identify mixed terms of the model, since there would no longer be significant integration errors. Than, the algorithm itself could be maybe improved. An idea is to consider weight factors inside the cost function. Indeed, depending on whether the terms of mass or drag have to be identified, measures of velocity or acceleration could be more or less significant.

In conclusion, the work I did for the "MDM Team" during my master thesis led to results that have been successfully applied to improve the functionality of Zeno. Moreover, the code I wrote could be re-used with the aim of identifying the model of new vehicles.

Appendix A

Matlab Library and GUI

All the code necessary to implement the identification procedure has been developed within MATLAB environment. The code has been divided into libraries containing classes and interfaces, with a strong Object-Oriented imprint. All the code is written using almost basic functions. In this way, the transposition into other programming language is much simpler and faster. Moreover, reducing the usage of MATLAB toolbox and high level functions, it is possible to have an higher control over the code.

A.1 Matlab Library

Without considering all the classes necessary for side tasks, like the porting of the data contained within .bag files into .mat files, two main libraries can be considered:

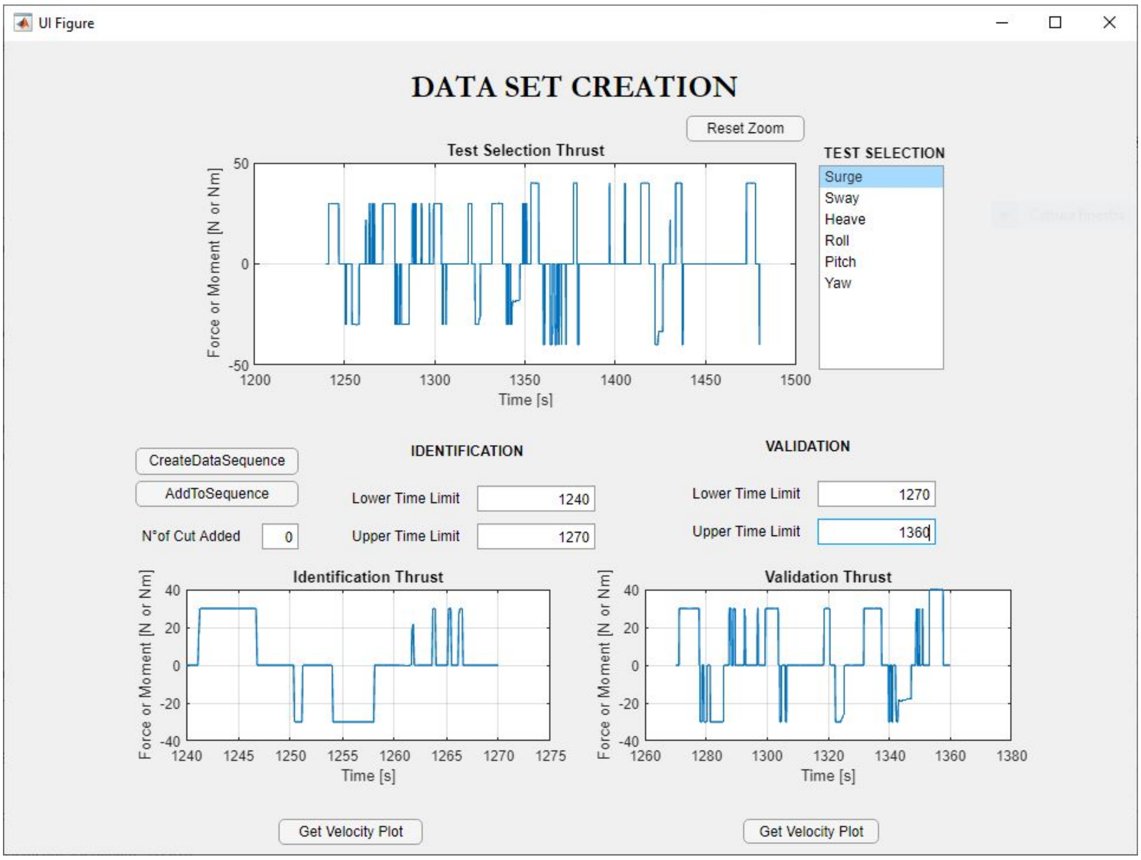
- A first library contains the classes necessary for the simulation of Zeno, used in the validation phase. The class that instantiate the concrete model of Zeno is a sub-class of a generic interface for the model simulation. This interface is part of the company's proprietary MATLAB libraries. The library contains also classes and interfaces for the simulation of the sensors mounted on the vehicle. All the necessary numerical parameters are placed in a single class in order to maximize the control on them.
- A second library contains the programs that implement the identification algorithms. As stated in chapter 3.4, all the algorithms are based on the same optimization problem, they just differs in the structure of the regressor form. For this reason, a common abstract interface is present. Several sub-classes are than present; each one has the task of instantiating an object that implements

a single optimization problem with its corresponding regressor. The form of each regressor have been found using a symbolic code converter designed by my supervisor [15].

A.2 MATLAB GUI

A MATLAB Graphic User Interface has been created, with the aim of unifying the process for the identification of the model in a single graphical program. Moreover, the use of a graphic interface allows people who are not confindent with MATLAB language to use it. Indeed, all the classes of the libraries necessary during the process are referred with the usage of callback functions linked to graphical entity like windows, buttons and plots.
The interface is distributed over two MATLAB apps:

- The first app permit the selection of desired data sets for the identification and the validation, chosen from larger data sets of measurements carried out during all tests performed for this purpose. It is also possible to combine data sets related to different time intervals in order to create a single identification or validation data set. This interface is shown in figure (A.1).
- The second app uses the data sets selected with the first one in order to perform a single identification and validation algorithm. The results of the identification are shown in the corresponding matrices and the validation is shown through the criteria and graphs mentioned in the chapter 4.1. This interface is shown in figure (A.2).



UI Figure

DATA SET CREATION

TEST SELECTION: Surge

IDENTIFICATION

Lower Time Limit 1240

Upper Time Limit 1270

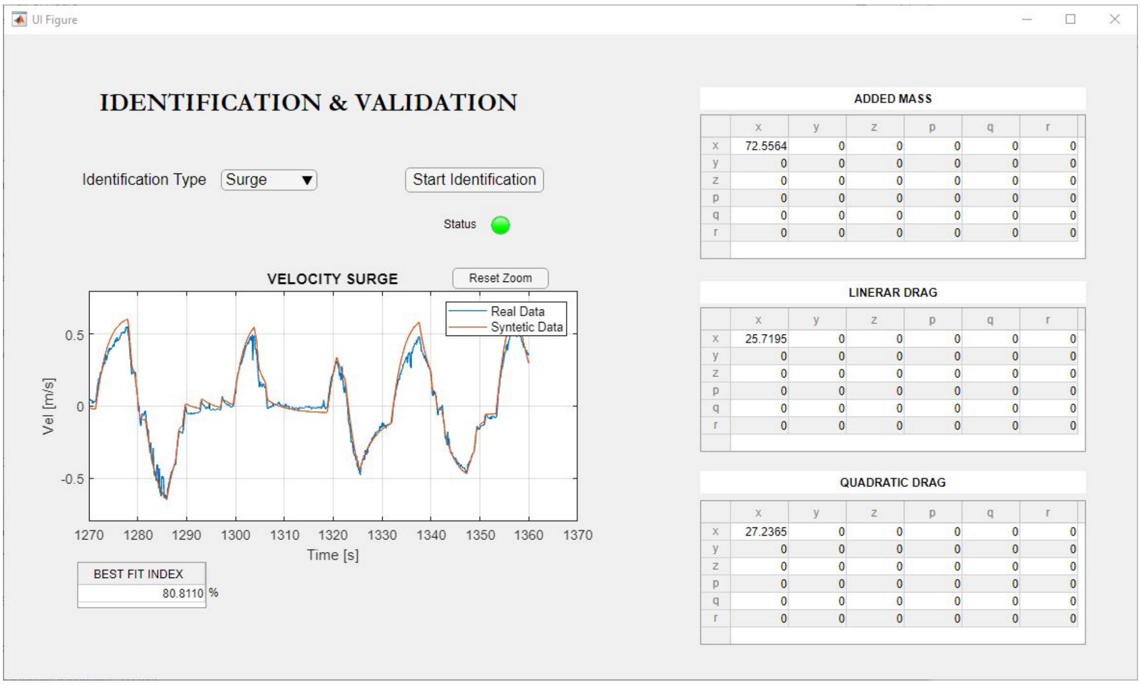
VALIDATION

Lower Time Limit 1270

Upper Time Limit 1360

N°of Cut Added 0

Figure A.1: Interface for the Selection of the Data Sets



IDENTIFICATION & VALIDATION

Identification Type: Surge

Status ●

BEST FIT INDEX: 80.8110 %

ADDED MASS

	x	y	z	p	q	r
x	72.5564	0	0	0	0	0
y	0	0	0	0	0	0
z	0	0	0	0	0	0
p	0	0	0	0	0	0
q	0	0	0	0	0	0
r	0	0	0	0	0	0

LINERAR DRAG

	x	y	z	p	q	r
x	25.7195	0	0	0	0	0
y	0	0	0	0	0	0
z	0	0	0	0	0	0
p	0	0	0	0	0	0
q	0	0	0	0	0	0
r	0	0	0	0	0	0

QUADRATIC DRAG

	x	y	z	p	q	r
x	27.2365	0	0	0	0	0
y	0	0	0	0	0	0
z	0	0	0	0	0	0
p	0	0	0	0	0	0
q	0	0	0	0	0	0
r	0	0	0	0	0	0

Figure A.2: Interface for the Identification and Validation of the Model Parameters

Bibliography

[1] W. Wang, R. Engelaar, X. Chen, and J. Chase, “The state-of-art of underwater vehicles - theories and applications,” 05 2009.

[2] [Online]. Available: https://www.kongsberg.com/maritime/products/marine-robotics/autonomous-underwater-vehicles/AUV-hugin/

[3] [Online]. Available: http://www.teledynemarine.com/gavia/

[4] B. Sahu and B. Subudhi, “The state of art of autonomous underwater vehicles in current and future decades,” pp. 1–6, 02 2014.

[5] [Online]. Available: https://www.mdmteam.eu/

[6] [Online]. Available: http://www.archeosub.eu/index.php/it/

[7] L. V. G. O. Bruno Siciliano, Lorenzo Sciavicco, *Robotics: Modelling, Planning and Control*. Springer Nature, 11 2008.

[8] T. I. Fossen, *Guidance and Control of Ocean Vehicles*. John Wiley & Sons Inc, 06 1994.

[9] [Online]. Available: http://www.ladispe.polito.it/corsi/mic/material/LEFSI_I.pdf

[10] L. E. G. Giuseppe Calafiore, *Optimization Models*. Cambridge University Press, 12 2014.

[11] L. V. Stephen Boyd, *Convex Optimization*. Cambridge University Press, 2004.

[12] [Online]. Available: http://cvxr.com/cvx/doc/intro.html\#what-is-cvx

[13] S. Bektas and Y. Sisman, “The comparison of l1 and l2-norm minimization methods,” *International Journal of the Physical Sciences*, vol. 5, pp. 1721–1727, 10 2010.

[14] [Online]. Available: http://www.ladispe.polito.it/corsi/mic/material/Lab4/Lab4.pdf

[15] [Online]. Available: https://www.cybertronics.cloud/