

UNIVERSIDADE FEDERAL DA BAHIA
DEE - DEPARTAMENTO DE ENGENHARIA ELÉTRICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

João da Costa Britto Neto

Parameters Identification of Dynamic Model and Simulation of Autonomous Underwater Vehicles

Dissertação de Mestrado

Salvador

2018

UNIVERSIDADE FEDERAL DA BAHIA
DEE - DEPARTAMENTO DE ENGENHARIA ELÉTRICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

João da Costa Britto Neto

Parameters Identification of Dynamic Model and Simulation of Autonomous Underwater Vehicles

Presented to the Master's Program in Electrical Engineering of the Federal University of Bahia in partial fulfillment of the requirements for the degree of Master of Science in Electrical Engineering.

Supervisor: Prof. Dr. André Gustavo Scolari Conceição

Salvador
2018

Ficha catalográfica elaborada pelo Sistema Universitário de Bibliotecas (SIBI/UFBA),
com os dados fornecidos pelo(a) autor(a).

da Costa Britto Neto, João
Parameters Identification of Dynamic Model and
Simulation of Autonomous Underwater Vehicles / João
da Costa Britto Neto. -- Salvador, 2018.
93 f. : il

Orientador: André Gustavo Scolari Conceição.
Dissertação (Mestrado - Programa de Pós-Graduação em
Engenharia Elétrica) -- Universidade Federal da
Bahia, Escola Politécnica, 2018.

1. Parameters Identification. 2. Underwater
Robotics. 3. Robot Simulator. 4. Gazebo. 5. Adaptive
Method. I. Scolari Conceição, André Gustavo. II.
Título.

Joao da Costa Britto Neto

Estimação de Parâmetros do Modelo Dinâmico e Simulação de Veículo Autônomo Submarino.

Dissertação apresentada à Universidade Federal da Bahia, como parte das exigências do Programa de Pós-Graduação em Engenharia Elétrica, para a obtenção do título de *Mestre*.

APROVADA em: 21 de Setembro de 2018.

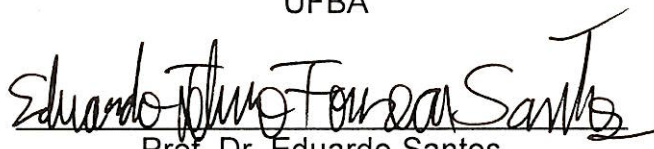
BANCA EXAMINADORA



Prof. Dr. André Gustavo Scolari Conceição
Orientador UFBA



Prof. Dr. Tiago Trindade Ribeiro
UFBA



Prof. Dr. Eduardo Santos
IFBA

Acknowledgements

I would like first to thank my adviser Prof. Dr. André Scolari, whose office's doors were always open whenever I had questions about my research or writing.

I give thanks to Shell for supporting the FlatFish project and to Brazilian National Agency of Petroleum (ANP) and Brazilian Industrial Research and Innovation Corporation (EMBRAPII) for funding this innovative project.

Thanks to the DFKI colleagues that somehow contribute to this work, specially Christopher Gaudig and Sascha Arnold for the aid and support with experiments at DFKI's facilities.

Many thanks to the BIR and the FlatFish team, those whose were always available to support each other. For those whose a mission given is a mission accomplished, even if it means working during weekends, holidays, in the evening, in a cold water tank or enduring seasickness in a boat upon a windy day, with the purpose of successfully performing FlatFish's tests. Working with the FlatFish team is a pleasant and enriching experience. The outcome of our work makes me proud by demonstrating that the development of highly innovative robots in Salvador, my homeland, is possible.

Many thanks to Marco Reis for inspiring the team in following our passions and overcome adversities, both in personal and professional life. Many thanks to Jan Albiez for guiding us in the underwater robotics field, for sharing his experience and believing in our potential since the beginning. Many thanks to Sylvain Joyeux for guiding us thought the field of software development, demonstrating by example how to develop elegant and robust software that enable ideas to become reality.

I would like also to thank my friends Diego César, Rafael Saback and Jessivaldo Santos for giving feedback and sharing thoughts on this work. Special thanks to Jessivaldo for proofreading this whole thesis.

Finally, I must express my gratitude to my parents, Sônia and João Claudio, and sisters, Mariana and Ana Carolina, for always encouraging me throughout my years of study. Last but not least, many thanks to my beloved Katharina for bringing joy to my life and providing with unfailing support and encouragement.

Thank you very much, everyone!

"Philosophy is written in that great book which ever lies before our eyes — I mean the universe — but we cannot understand it if we do not first learn the language and grasp the symbols, in which it is written. This book is written in the mathematical language, and the symbols are triangles, circles and other geometrical figures, without whose help it is impossible to comprehend a single word of it; without which one wanders in vain through a dark labyrinth."

Galileo Galilei

Resumo

Este trabalho trata sobre modelagem dinâmica, identificação de parâmetros e simulação de veículos submarinos. Simulação tem um papel importante no desenvolvimento de veículos submarinos. Entretanto, os simuladores robóticos comumente utilizados na atualidade, como o Gazebo, não permitem uma representação fidedigna da dinâmica de um veículo submarino, uma vez que os motores de física empregados no simulador não são capazes de representar o termo hidrodinâmico da massa adicional. Essa falta de representatividade não permite um cenário de teste realista do ambiente submarino. Este trabalho tem por objetivo propor uma abordagem matemática que visa sobrepor a incapacidade do simulador em representar a massa adicional sem a necessidade de alterar o motor de física.

Para simular o veículo submarino, é necessário saber os parâmetros do modelo dinâmico. Assim, o presente trabalho tem também por objetivo identificar os parâmetros do modelo dinâmico do veículo autônomo submarino FlatFish. Duas técnicas de identificação de parâmetros foram empregadas: método dos mínimos quadrados e identificador adaptativo. Na sequência foram realizados experimentos, em tanque e no mar, para identificar os parâmetros do modelo dinâmico simplificado do FlatFish, utilizando dados coletados experimentalmente dos sensores embarcados no veículo. Análises comparativas de velocidades entre dados medidos do veículo real e do modelo simulado atuado sobre os mesmos esforços de controle mostram que os parâmetros identificados por ambos os métodos são capazes de representar a dinâmica do veículo quando acionado em um grau de liberdade. No entanto verificou-se que os parâmetros do modelo simplificado não são capazes de representar a dinâmica do veículo quando atuado em vários graus de liberdade simultaneamente. Verificou-se que há a necessidade de realizar uma identificação do modelo completo para que a simulação do Gazebo juntamente com a técnica proposta para representar a massa adicional, seja capaz de emular o comportamento do veículo real. Apesar disso, os parâmetros identificados podem ser utilizados nos projetos de controladores baseados em modelo e em filtros estimadores de estado.

Palavras-chave: Identificação de Parâmetros, Mínimos Quadrados, Método Adaptativo, Robótica Submarina, Simulação Robótica, Gazebo

Abstract

This work address the topics of dynamical modeling, parameters identification and simulation of underwater vehicles. Simulation plays an important role in the development of underwater vehicles. Nevertheless, currently used robot simulators, such as Gazebo, do not allow an accurate representation of the underwater vehicle's dynamics, since the physic engine employed by the simulator is not able to represent the hydrodynamical term of the added mass. This lack of representativeness does not allow a realist test scenario of the underwater environment. This work has as objective to propose a mathematical approach to overcome the inability of the simulator in representing the added mass without changes to the physic engine.

To simulate the underwater vehicle, it is necessary to know the parameters of the dynamical model. Therefore, this work has also for objective to identify the parameters of the dynamical model of the autonomous underwater vehicle FlatFish. Two parameters identification techniques were applied: least square method and adaptive identifier. In sequence, experiments were conducted, in basin and in the sea, with the purpose of identifying the parameters of the simplified dynamical model of FlatFish, using data collect experimentally from vehicle's onboard sensors. Comparative analysis of velocities among vehicle's measured data and the simulated model actuated upon the same control effort show that the parameters identified by both methods are able to represent the dynamics of the vehicle when actuated in one degree of freedom. However, it was verified that the parameters of the simplified model are not able to represent the vehicle's dynamics when actuated in several degree of freedom simultaneously. It was established the need of performing parameters identification for the complete dynamical model so that the Gazebo simulation altogether with the proposed technique to represent the added mass, be able to emulate the behavior of the real vehicle. Despite that, the identified parameters can be used in the project of model based controllers and in state estimator filters.

Keywords: Parameters Identification, Least Square, Adaptive Method, Underwater Robotics, Robot Simulator, Gazebo

List of Figures

Figure 1 – FlatFish at the DFKI RIC test facilities in Bremen, Germany, [1]. . . .	20
Figure 2 – Configuration of FlatFish’s thrusters, [2].	22
Figure 3 – FlatFish’s rim-driven thruster, [3].	23
Figure 4 – Thruster coefficient.	23
Figure 5 – Control chain used in FlatFish, [4].	26
Figure 6 – Body frame of FlatFish [2].	27
Figure 7 – Schematic used to evaluate Gazebo Simulation.	39
Figure 8 – Method 1: Results of velocities and efforts expressed in body frame: (a) linear velocities, (b) forces, (c) angular velocities, (d) torques	40
Figure 9 – Method 2: Results of velocities and efforts expressed in body frame: (a) linear velocities, (b) forces, (c) angular velocities, (d) torques	41
Figure 10 – FlatFish performing experiments in test basin	48
Figure 11 – Cross validation in surge performed in basin: (a) Velocity, full experiment, (b) Velocity, 60s magnification, (c) Applied force, full experiment, (d) Applied force, 60s magnification	51
Figure 12 – Cross validation in sway performed in basin: (a) Velocity, full experiment, (b) Velocity, 60s magnification, (c) Applied force, full experiment, (d) Applied force, 60s magnification	52
Figure 13 – Cross validation in heave performed in basin: (a) Velocity, full experi- ment, (b) Velocity, 60s magnification, (c) Applied force, full experiment, (d) Applied force, 60s magnification	53
Figure 14 – Cross validation in yaw performed in basin: (a) Velocity, full experiment, (b) Velocity, 60s magnification, (c) Applied torque, full experiment, (d) Applied torque, 60s magnification	54
Figure 15 – Test site located at Marina de Aratu in Aratu bay (Google Maps) . . .	55
Figure 16 – Catamaran used as laboratory, docked at Marina de Aratu	56
Figure 17 – FlatFish preparing for experiments in bay	56
Figure 18 – Cross validation in surge performed in bay: (a) Velocity, full experiment, (b) Velocity, 60s magnification, (c) Applied force, full experiment, (d) Applied force, 60s magnification	58
Figure 19 – Cross validation in sway performed in bay: (a) Velocity, full experiment, (b) Velocity, 60s magnification, (c) Applied force, full experiment, (d) Applied force, 60s magnification	59
Figure 20 – Cross validation in heave performed in bay: (a) Velocity, full experiment, (b) Velocity, 60s magnification, (c) Applied force, full experiment, (d) Applied force, 60s magnification	60

Figure 21 – Cross validation in yaw performed in bay: (a) Velocity, full experiment, (b) Velocity, 60s magnification, (c) Applied torque, full experiment, (d) Applied torque, 60s magnification	61
Figure 22 – Multi DOF actuation in basin, linear DOF: (a) Velocity surge, (b) Force surge, (c) Velocity sway, (d) Force sway, (e) Velocity heave, (f) Force heave	64
Figure 23 – Multi DOF actuation in basin, angular DOF: (a) Angular velocity roll, (b) Torque roll, (c) Angular velocity pitch, (d) Torque pitch, (e) Angular velocity yaw, (f) Torque yaw	65
Figure 24 – Signal with noise and its first derivative computed by Savitzky-Golay filter	70
Figure 25 – Identification dataset for surge, done in basin: (a) Full trial, (b) 60s zoom in	71
Figure 26 – Output of adaptive identifier for surge. $a_m = 1, \lambda_1 = 2e - 4, \lambda_2 = 5e - 1, \lambda_3 = 5e - 1, \lambda_4 = 0$: (a) Parameters, (b) Velocity error	71
Figure 27 – Identification dataset for sway, done in basin: (a) Full trial, (b) 60s zoom in	72
Figure 28 – Output of adaptive identifier for sway. $a_m = 1, \lambda_1 = 2e - 4, \lambda_2 = 5e - 1, \lambda_3 = 5e - 1, \lambda_4 = 0$: (a) Parameters, (b) Velocity error	72
Figure 29 – Identification dataset for heave, done in basin: (a) Full trial, (b) 60s zoom in	73
Figure 30 – Output of adaptive identifier for heave. $a_m = 1, \lambda_1 = 1.5e - 4, \lambda_2 = 5e - 1, \lambda_3 = 5e - 1, \lambda_4 = 5e - 2$: (a) Parameters, (b) Velocity error	73
Figure 31 – Identification dataset for yaw, done in basin: (a) Full trial, (b) 60s zoom in	74
Figure 32 – Output of adaptive identifier for yaw. $a_m = 1, \lambda_1 = 1e - 2, \lambda_2 = 1e4, \lambda_3 = 5e0, \lambda_4 = 0$: (a) Parameters, (b) Velocity error	74
Figure 33 – Identification dataset for surge, done in bay: (a) Full trial, (b) 60s zoom in	75
Figure 34 – Output of adaptive identifier for surge. $a_m = 1, \lambda_1 = 2e - 4, \lambda_2 = 5e - 1, \lambda_3 = 5e - 1, \lambda_4 = 0$: (a) Parameters, (b) Velocity error	75
Figure 35 – Identification dataset for sway, done in bay: (a) Full trial, (b) 60s zoom in	76
Figure 36 – Output of adaptive identifier for sway. $a_m = 1, \lambda_1 = 2e - 4, \lambda_2 = 5e - 1, \lambda_3 = 5e - 1, \lambda_4 = 0$: (a) Parameters, (b) Velocity error	76
Figure 37 – Identification dataset for heave, done in bay: (a) Full trial, (b) 60s zoom in	77
Figure 38 – Output of adaptive identifier for heave. $a_m = 1, \lambda_1 = 1.5e - 4, \lambda_2 = 5e - 1, \lambda_3 = 5e - 1, \lambda_4 = 5e - 2$: (a) Parameters, (b) Velocity error	77
Figure 39 – Identification dataset for yaw, done in bay: (a) Full trial, (b) 60s zoom in	78
Figure 40 – Output of adaptive identifier for yaw. $a_m = 1, \lambda_1 = 5e - 4, \lambda_2 = 5e1, \lambda_3 = 5e - 1, \lambda_4 = 0$: (a) Parameters, (b) Velocity error	78

Figure 41 – Output over timer of adaptive identifier in surge DOF as function of λ_1 , $a = 0.1$: (a) Inertia (b) Linear Damping, (c) Quadratic Damping, (d) Velocity Error	80
Figure 42 – Output over timer of adaptive identifier in surge DOF as function of λ_1 , $a = 0.5$: (a) Inertia (b) Linear Damping, (c) Quadratic Damping, (d) Velocity Error	81
Figure 43 – Output over timer of adaptive identifier in surge DOF as function of λ_1 , $a = 1$: (a) Inertia (b) Linear Damping, (c) Quadratic Damping, (d) Velocity Error	82
Figure 44 – Output over timer of adaptive identifier in surge DOF as function of λ_1 , $a = 1.5$: (a) Inertia (b) Linear Damping, (c) Quadratic Damping, (d) Velocity Error	83
Figure 45 – Output over timer of adaptive identifier in surge DOF as function of λ_1 , $a = 2$: (a) Inertia (b) Linear Damping, (c) Quadratic Damping, (d) Velocity Error	84
Figure 46 – Variation of gain λ_1 in adaptive identifier, with gain $a = 0.1$: (a) Identified parameters, using identification dataset (b) Mean absolute error of identified parameters, using cross-validation dataset	84
Figure 47 – Variation of gain λ_1 in adaptive identifier, with gain $a = 0.5$: (a) Identified parameters, using identification dataset (b) Mean absolute error of identified parameters, using cross-validation dataset	85
Figure 48 – Variation of gain λ_1 in adaptive identifier, with gain $a = 1$: (a) Identified parameters, using identification dataset (b) Mean absolute error of identified parameters, using cross-validation dataset	85
Figure 49 – Variation of gain λ_1 in adaptive identifier, with gain $a = 1.5$: (a) Identified parameters, using identification dataset (b) Mean absolute error of identified parameters, using cross-validation dataset	86
Figure 50 – Variation of gain λ_1 in adaptive identifier, with gain $a = 2$: (a) Identified parameters, using identification dataset (b) Mean absolute error of identified parameters, using cross-validation dataset	86

List of Tables

Table 1 – FlatFish Specifications, [1]	21
Table 2 – Thruster Configuration Matrix for FlatFish	24
Table 3 – Definition of lumped parameters	32
Table 4 – Experimental Information	39
Table 5 – Parameters Identification Experiments	49
Table 6 – Parameters Cross-Validation Experiments	49
Table 7 – Adaptive Identifier Gains Used in Basin Dataset	49
Table 8 – Adaptive and Least-Squares Parameters, Basin Dataset	50
Table 9 – Parameters Identification Experiments	55
Table 10 – Parameters Cross-Validation Experiments	55
Table 11 – Adaptive Identifier Gains Used in Bay Dataset	57
Table 12 – Adaptive and Least-Squares Parameters, Bay Dataset	57
Table 13 – Multi DOF Experiment	60
Table 14 – Mean Absolute Error in Multi DOF Experiment	63
Table 15 – Parameters of signal with noise	69
Table 16 – Savitzky-Golay filter’s kernel, polynomial order 3, 33 samples and first derivative at the center point	70

List of abbreviations and acronyms

ANP	Brazilian National Agency of Petroleum
API	Application Program Interface
AUV	Autonomous Underwater Vehicle
BIBS	Bounded-Input Bounded-State
CFD	Computational Fluid Dynamics
COG	Center of Gravity
DFKI	German Research Center for Artificial Intelligence
DOF	Degree of Freedom
DVL	Doppler Velocity Log
EMBRAPII	Brazilian Industrial Research and Innovation Corporation
FIR	Finite Impulse Response
HIL	Hardware-In-the-Loop
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
KF	Kalman Filter
LCP	Linear Complementary Problem
MAE	Mean Absolute Error
MSS	Marine System Simulator
ODE	Open Dynamic Engine
RIC	Robotics Innovation Center
ROCK	Robot Construction Kit
ROV	Remotely Operated Vehicle
SDF	Simulation Description Format
SPD	Symmetric Positive Definite

TCM	Thruster Configuration Matrix
USBL	Ultra-short Baseline
UUV	Unmanned Underwater Vehicle

Contents

1	INTRODUCTION	16
1.1	Motivation	16
1.2	General Objective	18
1.3	Specific Objectives	18
1.4	Thesis Outline	19
2	FLATFISH	20
2.1	Sensors	21
2.1.1	Navigation System	21
2.1.2	Inspection System	22
2.2	Propulsion	22
2.2.1	Thruster Model	22
2.2.2	Thruster Configuration Matrix	24
2.3	Software Architecture	25
2.4	Control Chain	25
3	DYNAMICAL MODELING OF UNDERWATER VEHICLE	27
3.1	Reference Frames	27
3.2	Kinematic Model	28
3.3	Dynamic Model of a Rigid Body	28
3.3.1	Hydrodynamic Effects	29
3.3.1.1	Hydrodynamic Damping	29
3.4	Complete Dynamic Model	30
3.5	Simplified Dynamic Model	30
4	SIMULATION OF UNDERWATER VEHICLE	33
4.1	Literature Review	33
4.2	Simulator Improvements for Underwater Vehicle Dynamics	35
4.2.1	Implementation in Gazebo	36
4.2.2	Simulation Procedures	38
4.2.2.1	Results	39
5	PARAMETERS IDENTIFICATION FOR UNDERWATER VEHICLES	42
5.1	Literature Review	42
5.2	1-DOF Model Identification	44
5.2.1	Least-Squares Parameters Identification	44
5.2.2	Adaptive Parameters Identification	44

6	EXPERIMENTAL RESULTS	47
6.1	Experiments in Basin	47
6.1.1	Experimental Setup	47
6.1.2	Results	48
6.2	Experiments in Bay	51
6.2.1	Experimental Setup	51
6.2.2	Results	53
6.3	Multi DOF Actuation	58
6.3.1	Experimental Setup	59
6.3.2	Results	62
7	CONCLUSION	66
7.1	Future works	67
A	SAVITZKY-GOLAY FILTER	68
B	PARAMETERS IDENTIFICATION DATASET	71
B.1	Basin	71
B.1.1	Surge	71
B.1.2	Sway	72
B.1.3	Heave	73
B.1.4	Yaw	74
B.2	Bay	75
B.2.1	Surge	75
B.2.2	Sway	76
B.2.3	Heave	77
B.2.4	Yaw	78
C	INFLUENCE OF GAIN IN ADAPTIVE IDENTIFIER	79
	BIBLIOGRAPHY	87

1 Introduction

1.1 Motivation

The oceans cover around 71% of Earth's surface. Of this total, about 95% are unexplored, unknown to man. As an example, more men stepped on the lunar surface than they arrived to the Mariana Trench, the deepest part of the world's oceans, [5].

The fact that this environment is inhospitable to the human presence is a key point so that the oceans remain unexplored. The use of Unmanned Underwater Vehicles (UUV) is crucial to overcome this obstacle. Those vehicles are capable of collecting data via embedded sensors that are used in several areas of human activities. UUVs are used in discovering and monitoring of marine life, [6], in shipwreck exploration, [7], in the inspection and maintenance of underwater structures, in mine countermeasures research, among others, [8].

UUVs can be classified in Remotly Operated Vehicle (ROV) and Autonomous Underwater Vehicle (AUV). ROVs, firstly developed, are tethered vehicles that requires human operator. Work class ROVs, used in deep water applications, usually have high operational costs associated with the dedicated support vessel required to accommodate the vehicle and its accessories, like the launch and recovery system and the tether management system, as well as the personal required to operate and maintain the ROV onboard, [9].

AUVs are untethered vehicles that carry their own power source and are able to navigate autonomously, without direct intervention of an human operator. This capability is of great interest to the offshore Oil & Gas industry, which requires frequent inspection of subsea structures, like pipelines, manifolds, etc. The use of non-resident AUVs allows a reduction of operational cost to the inspection missions, since it simplifies the operation in terms of vessel requirements. For an AUV application, there is no need to perform tether management nor to have a control room in the vessel, which is used basically for launching and recovering the vehicle, [10].

The goal of the Oil & Gas industry is to perform on-demand inspection and support surveillance operation in remote areas without the need of deploying a dedicated vessel, by using subsea-resident AUVs. Such an AUV is assisted by a subsea docking station, which is used for battery recharging and data transfer of mission plans and inspection results, [10], [11]. Projects like Sabertooth, [12], and FlatFish, [1], are examples of vehicles that aim to archive this goal.

The development of AUVs, specially those envisioned to be subsea residents, is a costly activity. Underwater vehicles are usually composed of expensive hardware and their

development requires several tests in marine environmental and/or large tanks during the process. With the purpose of reducing costs and risks associated with those operations, it is usual the deployment of simulated environment to test new concepts, strategies and algorithms. Controllers, state estimators, navigation and perception are example of functionalities that can be tested in simulation.

It is desirable the simulation to be as faithful as possible to the real physical world, i.e. the dynamical behavior of simulated vehicle to be analogous to the real vehicle's behavior. This characteristic is important for the test of controllers for example.

Currently, 3D robot simulator such as Gazebo, [13], becomes an essential tool to roboticists. It makes use of an embedded physics engine to simulate rigid-body dynamics and collision, allowing a reliable simulation of aerial and ground vehicles. Nevertheless, the same can not be said of underwater robots. The complexity of hydrodynamic effects acting on an underwater vehicle is not fully incorporated in standard physics engines. Some works either consider only the simpler hydrodynamics effects such as buoyancy and damping, [14], or implement custom solution to include the challenging effects caused by the added mass, [15], [16]. Despite those solutions, the appropriate method to incorporate the added mass on standard robot simulator, including its physics engine, is still an open issue.

Other key point to obtain a reliable simulation is to determine the correct dynamical parameters of the virtual vehicle. The parameter identification of the dynamical model is usually a complex and operationally demanding task, due the non-linear and coupled characteristic of UUVs. Usually, parameters identification of UUVs is performed with experimental data collected from onboard sensor, using identification techniques performed either online or offline, [17], [18], [19], [20].

Being able to identify the parameters of the dynamical model is of benefit not only for the quality of the simulation environment. Model-based controllers, state estimators and model-based fault detection and diagnosis are equally enhanced by the incorporation of the dynamical model during UUV operation. [4] applied a nonlinear model predictive control to the FlatFish AUV. [21] employed several model-based controllers to a ROV and compared their performances, concluding that model-based controllers outperformed the proportional derivative controller in trajectory tracking.

For the AUV case, the necessity of the dynamics model is even more relevant for state estimation and navigation. Underwater vehicles can not rely in sensors or communication based on electromagnetic waves, such as GPS, WiFi and communication via satellite, due the strong attenuation caused by sea water, [22]. Usually, AUVs have at their disposal sensors based on acoustic waves, like Doppler Velocity Log (DVL) for the measurement of velocity and Ultra-Short Baseline (USBL) or array of Long Baseline (LBL) for the measurement of position. Nevertheless the acoustic sensors can have a high latency or even have discarded samples.

The common way of determining the position of an UUV is by dead-reckoning with DVL and Inertial Navigation System (INS) data, i.e. by integrating the linear and angular velocities, what causes the error to increase continuously, [23].

It is common to use the dynamical model of the vehicle to aid the navigation system. [24] applied a model-aided INS with a Kalman filter to increase navigation performance of an AUV deployed in ocean. [25] made use of a nonlinear dynamic model-based state estimator to a ROV, both in laboratory and in-field, comparing the nonlinear observer with an extended Kalman filter.

For the case of fault detection, many works explores the importance of a model-based implementation. [26] explores some techniques of model-based residual generation using parameters identification and state estimation methods. Residuals are functions that are accentuated by faults, being a simple example the difference between the measured output and the output provided by the nominal model. [27] applied the residual analyses in order to detect thruster fault, by comparing measured data with the estimation provided by vehicle's model. [28] explores the possibility to do fault detection by performing online parameters identification and comparing the identified values with nominal ones known a priori. It is based on the assumption that faults are reflected in the physical characteristic of the system, by altering the parameters of the model.

The efficiency of such controllers, state estimators and fault detectors is subjected to the availability of model's parameters that is able to represent the vehicles dynamics.

1.2 General Objective

The main objective of this work is to obtain the parameters of the dynamical model of an AUV via adaptive technique and least square method, using experimental data collected from onboard sensors. The purpose of the identified parameters is to assess the simulated vehicle in the robot simulator Gazebo and in a custom simulator, with relation to logged data from the real vehicle.

1.3 Specific Objectives

- Develop algorithm to simulate the dynamics of an AUV;
- Improve robotic simulator Gazebo to consider the added mass effect present in underwater dynamics;
- Identify parameters of the simplified dynamical model of an AUV;
- Analyze performance of the adaptive identifier and of the least square method;

- Analyze the influence of gains in the adaptive identifier;
- Analyze the behavior of a simulated vehicle with simplified dynamics when performing a multi degree of freedom actuation;

1.4 Thesis Outline

This document is comprised by this chapter, that presents the importance of simulation for the development of underwater robotics and its requirement of the model's parameters, the other areas in which the dynamics model parameters are used, the motivation and goals of this work.

Chapter 2 presents the FlatFish AUV, the underwater vehicle used during experiments.

Chapter 3 introduces to the modeling of underwater vehicles, including the hydrodynamics terms and the simplification procedure commonly applied.

Chapter 4 explores the simulation tools for underwater vehicles, considering its particularities due the hydrodynamics influence.

Chapter 5 is dedicated to the parameters identification techniques of underwater vehicles, with a focus on the adaptive identifier and the least square method.

Chapter 6 presents the experiments of parameters identification of the FlatFish vehicle both in a basin and at the sea. At the end of the chapter, the identified parameters is analyzed within the Gazebo simulation for the multi degree of freedom actuation case.

Chapter 7 concludes the work and presents suggestions for future work.

Appendix A presents the Savitzky-Golay filter and its ability to derive a set of data based on least-square technique.

Appendix B shows the dataset used for parameters identification as well as the model's parameters provided by the adaptive identifier.

Appendix C explores the influence of gains in the adaptive identifier. Five hundreds sets of model's parameters were identified, each one with an unique combination of gains.

2 FlatFish

FlatFish, represented in Figure 1, is an autonomous underwater vehicle developed by the Brazilian Institute of Robotics (BIR) at SENAI CIMATEC located in Salvador, in partnership with the Robotics Innovation Center (RIC), the Bremen location of the German Research Center for Artificial Intelligence (DFKI). Two FlatFish vehicles were build as the result of a four years project for the Royal Dutch Shell, after it acquired BG group, with funds provided by the Brazilian National Agency of Petroleum (ANP) and the Brazilian Agency for Industrial Research and Innovation (EMBRAPPII).

The aim of FlatFish is to perform repeated inspection of Oil&Gas subsea structures, such as pipelines, manifolds and subsea isolation valve (SSIV). It was designed to be a subsea-resident AUV, meaning that a docking station present on the sea bottom enables the vehicle to recharge its battery and exchange data with a topside base while underwater.

A mission scenario for FlatFish is, [1]: The vehicle, in rest at the docking station, receives an inspection mission of a particular asset, e.g. a SSIV, from topside. FlatFish leaves the docking station and navigates heading to the designated structure, using external references, like pipelines, as guides. While in transit, the vehicle makes use of its perception sensors to avoid obstacles and to record data of the asset's status. Once it reaches the area of interest, visual and acoustic data is gathered from the inspection object. The vehicles than navigates back to the docking station using the same approach as before. In the docking station, the vehicles batteries are recharged and the logged data is downloaded to topside servers, where it can be processed.



Figure 1 – FlatFish at the DFKI RIC test facilities in Bremen, Germany, [1].

2.1 Sensors

FlatFish has advanced sensors that allows it to accomplish its goals: autonomous navigation in an subsea Oil&Gas field and inspection of its assets. Table 1 shows an overview of FlatFish’s features.

Table 1 – FlatFish Specifications, [1]

Depth rating	300 m
Weight (in air)	275 kg
Size (LWH)	220 cm x 105 cm x 50 cm
Propulsion	6x 60N Enitech ring thrusters (120N in each direction)
Battery	Lithium-Ion battery 5,8 kWh (11,6 kWh) @ 48V
Communication (surface)	Rock7mobile RockBlock Iridium satellite modem (1,6 GHz) Digi XBee-Pro-868 (868 MHz) Ubiquiti PicoStation M2 HP WLAN-Modul (2,4 GHz)
Communication (submerged)	Evologics S2CR 48/78 kHz usable as USBL transponder
Communication (tethered)	10 GBit/s optical fibre 1 GBit/s Cat5e (max. 50m)
Light	4x Bowtech LED-K-3200 (3200 lumen each)
Laser Line projector	2x Picotronic LD532-20-3(20x80)45-PL line laser 20mW each @ 532nm
Sonar	BlueView MB1350-45 Multibeam Profiler (inspection sonar) Tritech Gemini 720i Multibeam Imager (navigation sonar) 2x Tritech Micron Sonar (obstacle avoidance)
Camera	4x Basler ace acA2040-gc25 2048x2048 at 25 frames/s, colour, GigabitEthernet
Depth sensor	Paroscientific 8CDP700-I, sampling rate of $5Hz$
INS/AHRS	KVH 1750 IMU, sampling rate of $100Hz$
DVL	Rowe SeaProfilr DualFrequency 300/1200 kHz, sampling rate of $4Hz$

2.1.1 Navigation System

Underwater navigation and localization are major challenges for AUVs. In the absence of a reliable position measurement, FlatFish fuses depth, linear and angular velocities data, provided by pressure sensor, doppler velocity log and inertial navigation system respectively, to perform dead-reckoning. As the error associate with the dead-reckoning increases over time, FlatFish uses its visual and acoustic perception sensors to identify known structures in the seafloor, providing a known location to the navigation system which is used to update the dead-reckoning, significantly reducing the estimation error. An ultra-short baseline placed in the docking station is also able to localize FlatFish’s within $1km$ radius and to delivery the computed position to the navigation system via vehicle’s acoustic modem.

2.1.2 Inspection System

The main goal of FlatFish is to perform inspection of subsea infrastructure. To achieve its objective, FlatFish possess visual and acoustic systems which can overcome environmental challenges such as high turbidity. The visual system comprises two pairs of stereo cameras, one looking ahead and the other downward, two pairs of LEDs and two lasers line projectors used for 3D reconstruction. The acoustic system is composed by a multi-beam profile sonar that can be used in low visibility.

2.2 Propulsion

FlatFish has six propellers distributed in the vehicle according Figure 2. They are rim-driven thrusters, meaning that they do not have shaft and gearbox, as seen in Figure 3. It allows the propeller to be relatively symmetrical and consequently to provide a similar thruster output in both direction, reaching up to $60N$.

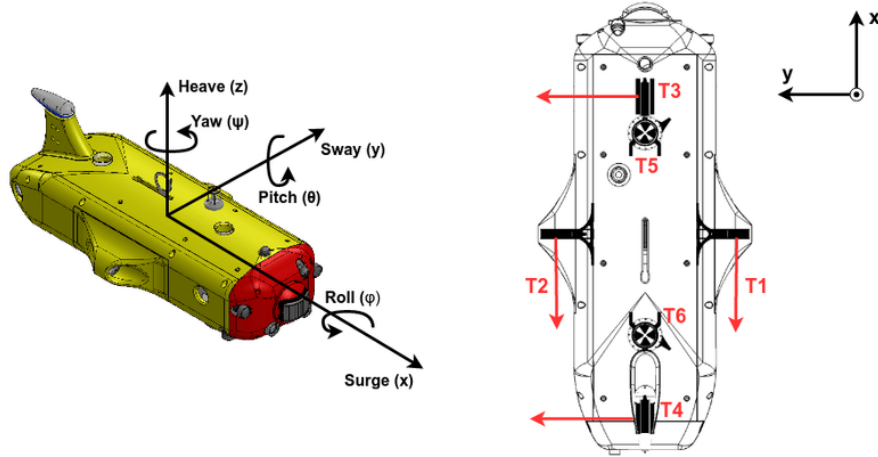


Figure 2 – Configuration of FlatFish's thrusters, [2].

2.2.1 Thruster Model

During vehicle's operation, the force applied by a thruster can not be direct measured, due to the complex interaction between thruster and water. Instead, its rotation speed is measured and used for controlling the propeller. Since the rotation speed is accessible, a common approach used to model the force applied by a thruster is given by, [29],

$$u_{tif} = C_t \Omega |\Omega|, \quad (2.1)$$

where $u_{tif} \in \mathbb{R}$ is the force applied by the thruster i , $C_t \in \mathbb{R}$ is the thruster coefficient and $\Omega \in \mathbb{R}$ is its rotation speed.



Figure 3 – FlatFish's rim-driven thruster, [3].

The coefficient C_t can be determined experimentally using least square method. Based on measured data of rotation speed and force given by the thruster's supplier, it is determined $C_t = 6.7 \times 10^{-5} N \cdot s^2 / rad^2$. Figure 4 shows the relation between thruster's force and rotation speed.

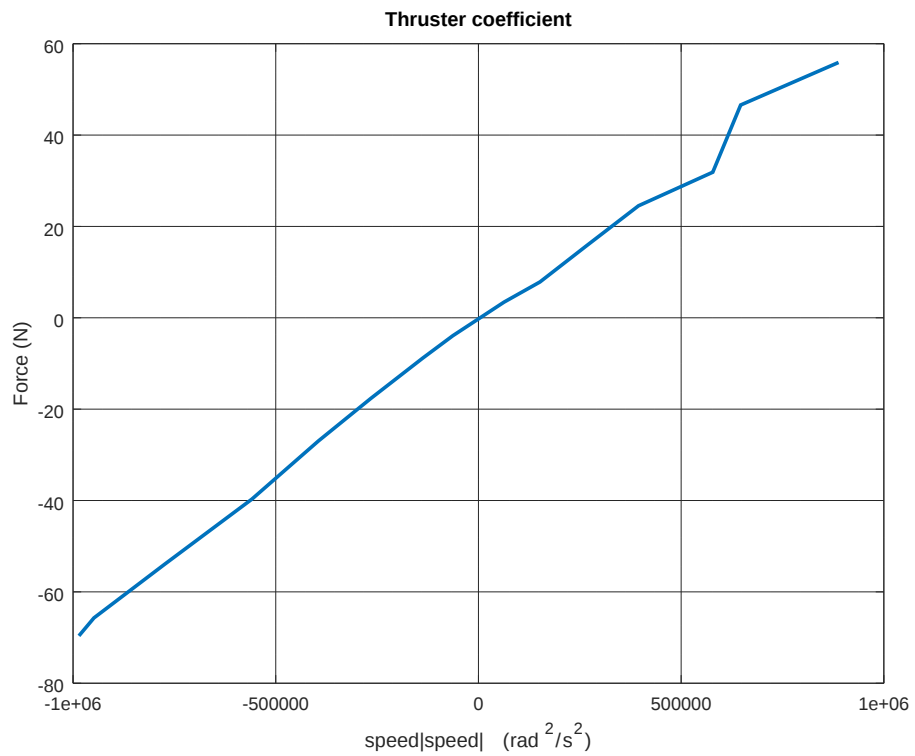


Figure 4 – Thruster coefficient.

2.2.2 Thruster Configuration Matrix

In order to determine the influence of a thruster in the vehicle, it is important to compute the force and torque applied by the propeller in the vehicle's center of gravity (COG). It requires a map of orientation and position of each thrusters, called Thruster Configuration Matrix (TCM), that convert the thruster's forces to vehicles forces and torques. The map is done as

$$u = Bu_{tf}, \quad (2.2)$$

where $u \in \mathbb{R}^6$ are the forces and torques applied in the vehicle by the thrusters, $B \in \mathbb{R}^{6 \times n}$ is the TCM, $u_{tf} \in \mathbb{R}^n$ is the force applied by each thruster and n is the amount of thrusters. In FlatFish case, $n = 6$.

Each column of the TCM can be computed as

$$B_i = \begin{bmatrix} R_{t_ib}e_1 \\ \mathcal{J}(p_{t_ib})R_{t_ib}e_1 \end{bmatrix}, \quad (2.3)$$

where

- $R_{t_ib} \in SO(3)$ is the rotation matrix from the thruster _{i} frame to the body frame,
- $p_{t_ib} \in \mathbb{R}^3$ is the position of the thruster _{i} frame in the body frame,
- $e_1 = [1 \ 0 \ 0]^T$ is the direction of propulsion in thruster _{i} frame, and
- $\mathcal{J} : \mathbb{R}^3 \rightarrow \mathbb{R}^{3 \times 3}$ is the skew-symmetric matrix used for cross product, given by

$$\mathcal{J} \left(\begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix} \right) = \begin{bmatrix} 0 & -p_3 & p_2 \\ p_3 & 0 & -p_1 \\ -p_2 & p_1 & 0 \end{bmatrix}. \quad (2.4)$$

For FlatFish, the identified TCM can be summarized as shown in Table 2.

Table 2 – Thruster Configuration Matrix for FlatFish

	T1	T2	T3	T4	T5	T6
Surge	1	1	0	0	0	0
Sway	0	0	-1	-1	0	0
Heave	0	0	0	0	1	1
Roll	0	0	0.032	-0.032	0	0
Pitch	-0.0315	-0.0315	0	0	-0.4235	0.556
Yaw	0.44	-0.4	-0.5735	0.936	0	0

2.3 Software Architecture

FlatFish’s software was implement using the Robot Construction Kit (Rock¹), a component-based framework for robotics development. It provides a rich set of tools and services required for developing, deploying and monitoring a robotic systems, such as visualization, logging, log replay, states monitoring, etc, [1].

The integration of software components that allows the creation of complex behaviors of the robot is done by Syskit². Syskit is the management layer responsible for establishing connection of component’s ports, applying specific configurations to the system, deploying the required components for a specific mission and monitoring the internal states of the system, [30].

A integration of Rock and the robot simulator Gazebo is available. It allows to test the integrated software stack and to verify the states transitions of the components on normal operation and during fault detection in a seamless way. Based on the architecture design used to integrate Rock-Syskit to Gazebo, the same software used in simulation can run on the real vehicle, requiring minimal changes for that.

2.4 Control Chain

FlatFish’s control chain is based on a P cascade controller, as shown in Figure 5. The designed control chain makes the assumption that the vehicle is passively stable in roll and pitch, since the vehicle’s center of buoyancy was designed to be above the center fo gravity. The outer controller is responsible for controlling the vehicles position and heading, which in turns provides the reference input for the velocities controller, what in sequence provides body efforts references.

The Buoyancy Compensation component includes the resultant gravitational term on the body efforts. From the perspective of the velocity controller, the vehicle have a neutral buoyancy. The total body efforts to be applied in the vehicle is converted in thruster’s force commands by the Optimal Allocation, which uses optimization theory for that purpose [2]. The thruster’s forces are then converted to set-point of rotation speed, according Equation 2.1, and are provided to vehicle’s thrusters.

FlatFish’s control chain has an operation frequency of $10Hz$.

¹ <http://rock-robotics.org/>

² <https://rock-core.github.io/rock-and-syskit/>

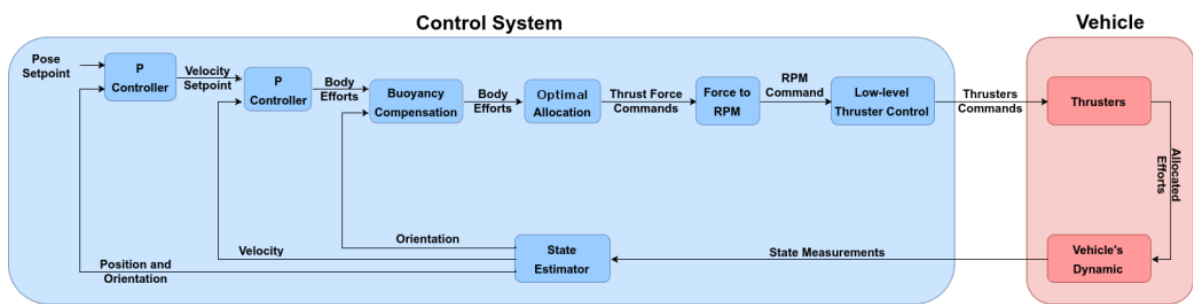


Figure 5 – Control chain used in FlatFish, [4].

3 Dynamical Modeling of Underwater Vehicle

Being able to compute the dynamical behavior of an underwater vehicle is an important factor for the operation of an AUV. The dynamical model can be used for state estimation, embedded in model based controllers or in simulation of the underwater vehicle. Determining the dynamical model is the first step in this process.

3.1 Reference Frames

First of all, it's required to establish the coordinates system to be used. In this work, the vehicle's body frame is according the Figure 6. The inertial reference frame is fixed in the water's surface, having the z axis pointing upward, [31]. It is worth noting that the reference system used is different of the one usually used by the naval community, where the z axis points downward, [32].

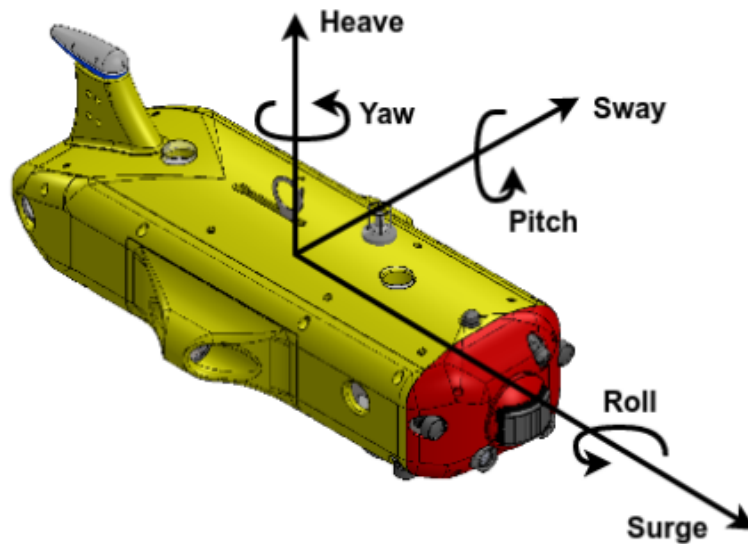


Figure 6 – Body frame of FlatFish [2].

3.2 Kinematic Model

The kinematics equation of a rigid body express in body frame is given by:

$$\begin{aligned}\dot{p} &= R\nu, \\ \dot{q} &= \frac{1}{2}q \cdot \begin{pmatrix} 0 \\ \omega \end{pmatrix},\end{aligned}\tag{3.1}$$

where

- $p \in \mathbb{R}^3$ is the position of the body frame's origin in the inertial frame,
- $R \in SO(3)$ is the rotation matrix from the body frame to the inertial frame,
- $q \in \mathbb{R}^4$ is the unitary quaternion representation of orientation R , with $q = \begin{pmatrix} q_0 \\ q_{1:3} \end{pmatrix}$ being $q_0 \in \mathbb{R}$ the real part and $q_{1:3} \in \mathbb{R}^3$ the imaginary part of the quaternion,
- $(\cdot)$ denotes quaternion's multiplication operator, [33], [34], and
- $\nu, \omega \in \mathbb{R}^3$ are, respectively, rigid body's linear and angular velocities, represented in body frame.

3.3 Dynamic Model of a Rigid Body

The dynamics equation of a rigid body with respects to the body frame can be represented in terms of body frame velocity, $v = [\nu^T \ \omega^T]^T$, and external efforts, $u_{RB} \in \mathbb{R}^6$, as [32]:

$$M_{RB}\dot{v} - \mathcal{H}(M_{RB}v)v = u_{RB},\tag{3.2}$$

where

$$M_{RB} = \begin{bmatrix} mI_{3 \times 3} & -m\mathcal{J}(c_G) \\ m\mathcal{J}(c_G) & I_0 \end{bmatrix}\tag{3.3}$$

represents the rigid body inertia matrix. Here,

- $m \in \mathbb{R}_+$ is the body's mass,
- $I_0 \in \mathbb{R}^{3 \times 3}$, $I_0 = I_0^T \succ \mathbf{0}$, is the inertia tensor with respect of the origin,
- $c_G \in \mathbb{R}^3$ is the center of gravity, and
- $I_{3 \times 3}$ is the identity matrix.

It made use of the skew-symmetric matrix $\mathcal{J} : \mathbb{R}^3 \rightarrow \mathbb{R}^{3 \times 3}$, defined in Equation 2.4. The term $\mathcal{H}(M_{RB}v)v$ represents the Coriolis and centripetal term. It can be parameterized such that $\mathcal{H}(M_{RB}v) = -\mathcal{H}(M_{RB}v)^T$, [32], given the function $\mathcal{H} : \mathbb{R}^6 \rightarrow \mathbb{R}^{6 \times 6}$ such as

$$\mathcal{H} \left(\begin{bmatrix} \nu \\ \omega \end{bmatrix} \right) = \begin{bmatrix} 0_{3 \times 3} & \mathcal{J}(\nu) \\ \mathcal{J}(\nu) & \mathcal{J}(\omega) \end{bmatrix}. \quad (3.4)$$

3.3.1 Hydrodynamic Effects

When moving in a fluid, the rigid body's dynamics is affected by the fluid that surround it. It's well understood the effect of restoring efforts due the Archimede's principle and the presence of damping that dissipate energy from the system. Other important term to be considered is the presence of kinetic energy in the fluid. Once the vehicle navigates, the fluid must move aside and close in behind the vehicle. As an consequence the fluid possess kinetic energy and any change on it need to be impart by the vehicle. This effect is the so called added mass, [32], [35].

The hydrodynamics effects acting on an underwater vehicle is given by, [32]:

$$u_H = -M_A \dot{v} + \mathcal{H}(M_A v)v - D(v)v + \mathcal{G}(R), \quad (3.5)$$

where

- $M_A \in \mathbb{R}^{6 \times 6}$ is the symmetric positive definite (SPD) matrix of hydrodynamic added mass,
- $\mathcal{H}(M_A v)$ represents the Coriolis and centripetal term of the added mass,
- $D(v) \in \mathbb{R}^{6 \times 6}$ is the hydrodynamic damping as a function of the velocity and is dissipative ($D(v) \succ \mathbf{0}$), and
- $\mathcal{G}(R)$ is the restoring forces and torques due gravity, such that $\mathcal{G}(R) = \mathcal{G}_0(R) + \mathcal{G}_b(R)$, with $\mathcal{G}_0(R) = - \begin{bmatrix} WR^T e_3 \\ W \mathcal{J}(c_G) R^T e_3 \end{bmatrix}$, $\mathcal{G}_b(R) = \begin{bmatrix} BR^T e_3 \\ B \mathcal{J}(c_B) R^T e_3 \end{bmatrix}$, being $W, B \in \mathbb{R}_+$ the weight and buoyancy of the vehicle, $c_G, c_B \in \mathbb{R}^3$ the center of gravity and center of buoyancy respectively and $e_3 = [0 \ 0 \ 1]^T$.

3.3.1.1 Hydrodynamic Damping

The representation of $D(v)$ is not definitive. The exact modeling of the hydrodynamic damping is hard to be obtained due the complexity of the several terms involved, like skin friction, vortex shedding, potential damping and wave drift damping, [32].

A general 6 DOF representation of the damping term considers a linear and a quadratic term, [32], [36]:

$$D(v) = D_L + \left(\sum_1^6 |v_i| D_{Q_i} \right), \quad (3.6)$$

where D_L and $D_{Qi} \in \mathbb{R}^{6 \times 6}$, whit $i = 1 \cdots 6$ referring to each DOF.

Previous studies show that for a coupling 6 DOF dynamics, the damping effect can be modeled using the fully coupled quadratic term, ignoring the linear contribution [37], [20], [38], such as

$$D(v) = \left(\sum_1^6 |v_i| D_{Qi} \right). \quad (3.7)$$

Other approach simplifies the damping effect. It consider the vehicle performing non-coupled motion, leading to uncoupled damping terms, [32], which results in

$$D(v) = \text{diag}\{d_{L_1}, d_{L_2}, d_{L_3}, d_{L_4}, d_{L_5}, d_{L_6}\} + \text{diag}\{|v_1|d_{Q_1}, |v_2|d_{Q_2}, |v_3|d_{Q_3}, |v_4|d_{Q_4}, |v_5|d_{Q_5}, |v_6|d_{Q_6}\}. \quad (3.8)$$

3.4 Complete Dynamic Model

Knowing that the efforts acting in an underwater vehicle are the control input and the hydrodynamical efforts,

$$u_{RB} = u_H + u, \quad (3.9)$$

the complete dynamical equation of an underwater vehicle can be obtained from Equations 3.2 and 3.5,[32], [20]:

$$M\dot{v} = \mathcal{H}(Mv)v - D(v)v + \mathcal{G}(R) + u, \quad (3.10)$$

where

- $M = M_{RB} + M_A$ is the inertia matrix
- $\mathcal{H}(Mv)$ represents the Coriolis and centripetal term,
- $D(v)$ is the damping effect,
- $u \in \mathbb{R}^6$ represent the forces and torques applied in the body frame,
- $v = [\nu^T \ \omega^T]^T$ is body frame's velocity, and
- $\mathcal{G}(R)$ represents the restoring forces and torques.

3.5 Simplified Dynamic Model

The 6 DOF dynamics equation of an underwater vehicle can be simplified, as the follow considerations are valid for some classes of UUV, [39],[17], [32]:

- The center of mass is located at the origin of the body frame,

- The body axis coincide with the principal axes of inertia, such that I_0 is diagonal,
- The vehicle have three plane of symmetry,
- The vehicle operates at small velocities ($< 2m/s$),
- The vehicle performs non-coupled motions, and
- The vehicle is undergoing small attitude motion.

Those assumptions are made in order to ignore the off-diagonal elements of matrices M , D_L and D_Q , to neglect the Coriolis and centripetal term, and to ignore movement in pitch and roll, which result in

$$\begin{aligned}
 & \begin{bmatrix} m_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & m_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & m_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & m_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & m_5 & 0 \\ 0 & 0 & 0 & 0 & 0 & m_6 \end{bmatrix} \dot{v} = \\
 & - \begin{bmatrix} d_{Q_1}|v_1| & 0 & 0 & 0 & 0 & 0 \\ 0 & d_{Q_2}|v_2| & 0 & 0 & 0 & 0 \\ 0 & 0 & d_{Q_3}|v_3| & 0 & 0 & 0 \\ 0 & 0 & 0 & d_{Q_4}|v_4| & 0 & 0 \\ 0 & 0 & 0 & 0 & d_{Q_5}|v_5| & 0 \\ 0 & 0 & 0 & 0 & 0 & d_{Q_6}|v_6| \end{bmatrix} v \\
 & - \begin{bmatrix} d_{L_1} & 0 & 0 & 0 & 0 & 0 \\ 0 & d_{L_2} & 0 & 0 & 0 & 0 \\ 0 & 0 & d_{L_3} & 0 & 0 & 0 \\ 0 & 0 & 0 & d_{L_4} & 0 & 0 \\ 0 & 0 & 0 & 0 & d_{L_5} & 0 \\ 0 & 0 & 0 & 0 & 0 & d_{L_6} \end{bmatrix} v + \begin{bmatrix} 0 \\ 0 \\ b_3 \\ 0 \\ 0 \\ 0 \end{bmatrix} + u. \quad (3.11)
 \end{aligned}$$

As such, the dynamical model equation of a decoupled single DOF of an underwater vehicle is given by ,[18], [40]:

$$m_i \dot{v}_i = -d_{Q_i} v_i |v_i| - d_{L_i} v_i + b_i + u_i, \quad (3.12)$$

where, for each degree of freedom i , m_i , d_{Q_i} , $d_{L_i} \in \mathbb{R}_+$ represent the total mass, the quadratic damping and the linear damping, respectively, $b_i \in \mathbb{R}$, $b_i = 0$ for $i \neq 3$, represents the resultant buoyancy and u_i is the control input.

Isolating the acceleration, Equation 3.12 becomes

$$\dot{v}_i = \alpha_i u_i + \beta_i v_i |v_i| + \mu_i v_i + \nu_i \quad (3.13)$$

Table 3 – Definition of lumped parameters

Lumped Parame- ters	Physical Definition
α_i	m_i^{-1}
β_i	$-m_i^{-1}d_{Q_i}$
μ_i	$-m_i^{-1}d_{L_i}$
ν_i	$m_i^{-1}b_i$

and can be written in the vectorial form

$$\dot{v}_i = \Phi_i^T f_i, \quad (3.14)$$

where $\Phi_i = [\alpha_i \ \beta_i \ \mu_i \ \nu_i]^T$ is the vector of lumped model parameters and $f_i = [u_i \ v_i|v_i| \ v_i \ 1]^T$ is a nonlinear vector of state and control input.

The convention between lumped parameters and their physical definition can be seeing in Table 3.

4 Simulation of Underwater Vehicle

This chapter explores the simulation of underwater vehicles, considering its particularities due the hydrodynamics influence. Section 4.1 performs a review on several simulators for underwater vehicles, both on the marine and the robotics field. It is given a focus on the Gazebo simulator and its integration with Rock framework, [41], since it's the system used during the FlatFish project. Section 4.2 proposes a mathematical approach to include the added mass effect on Gazebo simulator and in the sequence it compare two implementation methods to enhance the Rock-Gazebo integration available.

4.1 Literature Review

The development of unmanned underwater vehicles usually requires validation of software in a simulation environment. Tests in virtual environments help reduce the high costs of deployment of a real UUV, that comes either by the necessity of a vessel for open water operation or by rely on the availability of a large enough tank. Moreover, simulation allows to test the vehicle in varied situations, detecting possible control and software errors prior to real deployment. However, despite the presence of virtual UUVs, model based control algorithms could not be extensively tested on simulation due the incapacity of off-the-shelf robot simulators to compute the interaction of a fluid with a submerged body with intricate geometry.

Robot simulators, like Gazebo [13], allow to represent the dynamic environment where a robot can interact. They make use of a physic engine, a visualization layer and an Application Program Interface (API) that enables the interaction between the simulator and third part software, which can be used for real-time Hardware-In-the-Loop (HIL) simulations.

Gazebo allows the integration with four physical engines, where the Open Dynamic Engine (ODE) [42] is the default option. The physics engine is responsible for simulating the dynamics and kinematics of articulated rigid-bodys, usually having a built-in collision detection. It does not consider fluid interaction or hydrodynamic effects.

Commercial HIL simulators for the off-shore industry are available, e.g. the CyberSea Simulator [43], [44]. This simulator is able to simulate the hydrodynamic effects of vessels as well as power and propulsion systems.

Previous works have developed underwater simulations to test UUV. The Marine Systems Simulator (MSS) [45] is a Matlab/Simulink platform that allows the simulation of marine systems like vessels and UUVs. It considers hydrodynamic effects and environmental

conditions like waves, wind and currents, allowing Simulink based controllers to be tested.

In [14], an UUV with a manipulator is simulated in Gazebo together with the ROS¹ framework, although buoyancy and damping effects are present, the added mass effect is not taken into account. In [46], a simulation environment for marine robotics called MARS is proposed. It enables the simulation of multiples UUVs, with a special focus on sonar imaging. Although several underwater related terms are present, like water current, waves, computation of the center of buoyancy and volume calculation, the added mass influence is not mentioned.

In [15], a ROS based simulation using Gazebo is developed, being able to simulate several UUVs. It addresses the problem of added mass in simulation, affirming that it can lead to instability depending on the relations of added mass matrix coefficients and vehicle's mass. The solution used in [15] to overcome this problem is to use a low-pass filter in the acceleration data to prevent an unstable simulation, but no performance results were shown.

The UWSim [16], a software tool for visualization and simulation of robotic underwater missions allows the added mass effect to be take into account. For that, it relies on a separate module written in Matlab that computes the UUV's states and update the vehicles pose in the scene. In parallel, it also makes use of the physics engine Bullet to handle contact interaction. In this case the Matlab module is used to overcome the inability of the physics engine to consider the added mass effect.

In [41], the Rock framework is integrated with Gazebo, allowing drop-in real-time simulations. The simulation resources are exported to the Rock system by instantiating and synchronizing Rock's components inside a Gazebo system plugin. The visualization of the simulation uses Rock's graphical user interface (GUI) packages to represent 3D data model, which is based on the 3D graphics toolkit named OpenSceneGraph. An AUV simulation is presented, in which the hydrodynamic effect and thruster model are independent Gazebo plugins. The underwater plugin incorporating many fluid effects like buoyancy, damping and water current. Nevertheless added mass effect is not considered.

The Rock-Gazebo integration, [41], has been used for the FlatFish team as a development tool, allowing tests of new software components developed in Rock on the virtual FlatFish vehicle deployed in the Gazebo simulation. Due the representation lack of the added mass effect on it, there was the opportunity to improve the available tool by proposing a way to make the simulation take the added mass effect in account.

¹ <http://www.ros.org/>

4.2 Simulator Improvements for Underwater Vehicle Dynamics

This present work proposes a mathematical approach to include the added mass effect in Gazebo, without the need to change the physics engine. It will be implemented based on the Gazebo plugin developed in [41] and will have the added mass inertia as parameters.

The well known motion model equations that describe an UUV is given by Equation 3.10. ODE, the default physics engine of Gazebo, uses a Lagrange multiplier velocity based model to derive the equation of motion of a rigid body and also Linear Complementarity Problems (LCP) to consider contact and friction with other bodies, [42], [47]. Without loss of generality, Gazebo integrates the dynamics of a rigid body represented in body frame, obtained from the Newton-Euler formulation, as

$$M_{RB}\dot{v} = \mathcal{H}(M_{RB}v)v + \mathcal{G}_0(R) + u_g, \quad (4.1)$$

where $u_g \in \mathbb{R}^6$ represents the forces and torques applied to the vehicle.

The objective is to make the behavior of the rigid body in Gazebo as similar as possible to the submerged rigid body's dynamics, including the added mass effect. Moreover, this should be implemented as a Gazebo plugin, since the modification of Gazebo and the underlying physics engine would be a major undertaking (if at all possible).

A common approach in underwater simulation for robotic system takes into account the buoyancy and damping terms only, completely ignoring the added mass effect, as in [14], [41] and [46]. In [15], the added mass effect is considered, where u_g in Equation 4.1 is proposed as

$$u_g = -M_A\dot{v} + \mathcal{H}(M_Av)v - D(v)v + \mathcal{G}_b(R) + u. \quad (4.2)$$

Such an approach can lead to instability. The acceleration is required to compute the hydrodynamical efforts, $-M_A\dot{v}$, which are used by the physical engine to determine the new acceleration, velocity and pose. It means that only the previous computed acceleration is available, i.e. using the effort $-M_A\dot{v}[k-1]$ to determine the new acceleration from $M_{RB}\dot{v}[k]$. In such way, u_g , represented by Equation 4.2 turns Equation 4.1 stable only if $M_A \prec M_{RB}$, or alternatively $\|M_{RB}^{-1}M_A\|_2 < 1$.

The proposed solution in this work is to consider a compensated effort C such that the derivative of the velocity computed by Gazebo becomes equal to that of UUV's equation. Rewriting the dynamics of an UUV given by Equation 3.10 as

$$(M_{RB} + M_A)\dot{v} = F(v, R), \quad (4.3)$$

where

$$F(v, R) = \mathcal{H}(Mv)v - D(v)v + \mathcal{G}(R) + u \quad (4.4)$$

and using F instead of $F(v, R)$ from now on for simplification.

Considering u_g as

$$u_g = \mathcal{H}(M_A v) v - D(v) v + \mathcal{G}_b(R) + u + C \quad (4.5)$$

and knowing that $\mathcal{H}(Mv) = \mathcal{H}(M_{RB}v) + \mathcal{H}(M_A v)$ and $\mathcal{G}(R) = \mathcal{G}_0(R) + \mathcal{G}_b(R)$, the dynamics computed by Gazebo, given by Equation 4.1, can be rewrite as

$$M_{RB} \dot{v} = \mathcal{H}(Mv) v - D(v) v + \mathcal{G}(R) + u + C, \quad (4.6)$$

which in turn can uses Equation 4.4 to become

$$M_{RB} \dot{v} = F + C. \quad (4.7)$$

Combining Equations 4.3 and 4.7, gives

$$\dot{v} = (M_{RB} + M_A)^{-1} F = M_{RB}^{-1} (F + C), \quad (4.8)$$

resulting in

$$C = (M_{RB}(M_{RB} + M_A)^{-1} - \mathbb{I}) F, \quad (4.9)$$

where $\mathbb{I} \in \mathbb{R}^{6 \times 6}$ is the identity matrix. The compensated effort C , given by Equation 4.9, when applied in the Gazebo formulation, as Equation 4.6, results in the same UUV's dynamics, given by Equation 3.10. In other words, using the compensated effort C , defined as a function of the total efforts applied in the vehicle, makes Gazebo computes the desired $\dot{v}$ of an UUV, considering the added mass.

As M_A and M_{RB} are SPD, it implies the follow relations:

$$(M_{RB} + M_A) \succ M_{RB} \succ \mathbf{0}, \quad (4.10)$$

$$\mathbb{I} \succ M_{RB}(M_{RB} + M_A)^{-1} \succ \mathbf{0}, \quad (4.11)$$

$$\mathbf{0} \succ (M_{RB}(M_{RB} + M_A)^{-1} - \mathbb{I}) \succ -\mathbb{I}. \quad (4.12)$$

The fact that $\|(M_{RB}(M_{RB} + M_A)^{-1} - \mathbb{I})\|_2 < 1$ is indicative of solution stability.

4.2.1 Implementation in Gazebo

An UUV can be described in Gazebo as a set (model) of rigid bodies (links) fixed to a main body. The external efforts applied in each link are directly transmitted to the main link, in such way that the resultant efforts are applied in the model's COG. The mass matrix M_{RB} is also computed related to the COG. This approach was chosen once UUV model identification methods are used to define parameters for the whole vehicle [20], [37], [40].

Two implementation methods were proposed. Method 1 considers that the efforts, u , applied in the vehicle are not accessible in the present step and therefore the compensated effort, C , makes use of the previous efforts applied in the vehicle. Method 2 does assume that they are available, and modifies them to compensate for the added mass.

Method 1

Gazebo does not provide access to u , in Equation 4.6, in the current simulation iteration, i.e. no access to the control effort applied by thrusters nor efforts due contact interaction with others models and the world. However, Gazebo does provide methods to access the previous resultant efforts applied to a link. We therefore apply compensated efforts C based on the resultant effort of previous step. This implies that Equations 4.7 and 4.9 become

$$M_{RB}\dot{v}[k] = F[k] + C[k], \quad (4.13)$$

with

$$C[k] = (M_{RB}(M_{RB} + M_A)^{-1} - \mathbb{I})F[k-1], \quad (4.14)$$

where $F[k]$ is the body frame resultant effort applied in the COG during iteration k . Attention need to be taken when determining $F[k-1]$. Gazebo's API provides methods that return the net force and torque applied in the link's COG, meaning that they do not consider the Coriolis and centripetal term of the rigid body, $\mathcal{H}(M_{RB}v)$.

In order to compute $F[k-1]$, the net efforts in the main body (a link) provided by Gazebo's API should be considered in the model's COG, by taking in account the torque at the model's COG generated by the main body's force. In sequence, the Coriolis and centripetal term, $\mathcal{H}(M_{RB}v)$, shall be added to the resultant effort and, at last, the influence of the compensated effort of the previous step, $C[k-1]$, shall be removed. In such a way, the computed term $F[k-1]$ can be used to determine the new compensated effort $C[k]$.

Method 2

The assumption here is that there is no contact interaction with other elements in the scene. In our implementation, both thruster efforts and hydrodynamic effects are implemented within Gazebo plugins that we control, thus we have access to them before they are passed to Gazebo for integration. We can therefore edit the forces applied in the present iteration, instead of accessing them in the next iteration. In such way, the compensated effort, given by Equation 4.9, can be expanded directly in Equation 4.5 leading to

$$u_g = (M_{RB}(M_{RB} + M_A)^{-1} - \mathbb{I})(\mathcal{H}(M_{RB}v)v + \mathcal{G}_0(R)) + M_{RB}(M_{RB} + M_A)^{-1}(\mathcal{H}(M_A v)v - D(v)v + \mathcal{G}_b(R) + u), \quad (4.15)$$

where $(M_{RB}(M_{RB} + M_A)^{-1} - \mathbb{I})$ applies to the effects considered by Gazebo and $M_{RB}(M_{RB} + M_A)^{-1}$ applies to the hydrodynamic and thruster efforts.

4.2.2 Simulation Procedures

The proposed solution was implemented as an improvement of the FlatFish AUV simulation presented in [41], using an already developed framework-simulator platform. It allows to run algorithms implemented in the Rock framework in Gazebo's simulation environment. The simulated vehicle is a virtual representation of the FlatFish AUV, presented in Chapter 2. The vehicles existing control chain is used to generate the forces applied by the thrusters. The vehicles simulation model, represented in the Simulation Description Format (SDF)², describes the inertia term of all links that form the vehicle. As the links have a constant pose between them, the COG of the overall model can be computed.

With the model's COG known, as well as the orientation and position of the thrusters, the resulting forces and torques in the vehicle can be determined. The TCM embeds this information. It results in the map of thrusters forces to vehicle's efforts, as described in Equation 2.2

In order to compare the proposed solutions with each other, the C++ library *uwv_dynamic_model*³ was used. The library simulates the dynamics of a submerged rigid body by integrating Equations 3.1 and 3.10, using the fourth order Runge-Kutta method. A initial state of pose and velocity (usually zero) is determined for the rigid body. It is also possible to configure the dynamical equation to consider the Coriolis term for a coupled model, as Equation 3.10, or to ignore it for a decoupled and simplified model, as Equation 3.11. For the work present in this section the coupled model was used.

The schematic used is depicted on Figure 7. The Gazebo simulation provides state information feedback (pose and velocity) to the control chain. The control chain provides thruster commands which are used to feed the simulated vehicle both in Gazebo as well as in *uwv_dynamic_model*. The body frame velocities of both models are compared in order to evaluate the proposed methods.

The parameters of Equation 3.10 were the same in Gazebo and *uwv_dynamic_model*. The simulated vehicle has a mass of 350 Kg, moments of inertia I_x , I_y and I_z of 146, 36, 124 Kgm² respectively. The added mass matrix $M_A = \text{diag}([500 \ 700 \ 1200 \ 290 \ 300 \ 400])$. The damping is composed by a linear and a quadratic terms, in such a way that $D(v) = \text{diag}([50 \ 50 \ 50 \ 45 \ 45 \ 45]) + |v|^T \text{diag}([40 \ 40 \ 40 \ 35 \ 35 \ 35])$, and a resultant buoyancy of 18N applied 0.1m above COG. The parameters used in these simulations match the values identified for the FlatFish system in [40].

² <http://sdformat.org/>

³ https://github.com/rock-control/control-uwv_dynamic_model

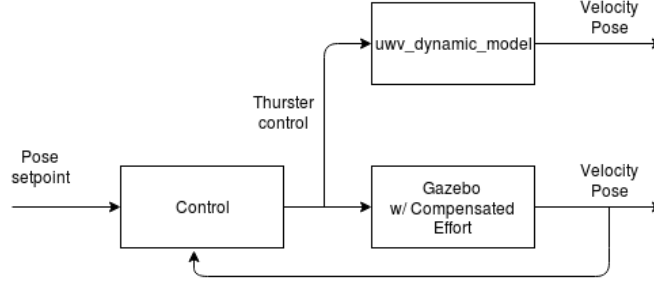


Figure 7 – Schematic used to evaluate Gazebo Simulation.

Table 4 – Experimental Information

Point	world_x	world_y	world_z	heading
Initial	0 m	0 m	-3 m	0°
Final	10 m	10 m	-8 m	180°

The initial depth is set to 3 m, to make sure the vehicle is underwater since *uwv_dynamic_model* does not take the surface in account. For each method proposed one simulation case is tested. In order to explore the complexity of the 6 DOF model, the simulation actuates 4 DOF simultaneously. The initial pose and setpoint pose are in Table 4. The control algorithm is not of interest in the present work, therefore its performance and the path to go from the initial to final point are not analyzed.

4.2.2.1 Results

Figures 8 and 9 show the results of simulations for methods 1 and 2 respectively.

For *Method 1*, shown in Figure 8, the simulation diverges significantly from *uwv_dynamic_model*. The angular movement is highly affected, leading the control chain to react abruptly, which in turn contributes to the oscillatory movement. One can notice that when the angular velocities are small the simulation and the expected model match. Other simulations indicate that this is also true if only one DOF is actuated.

For *Method 2*, shown in Figure 9, the velocities in simulation match the ones of *uwv_dynamic_model*. The compensation is done directly on the control efforts, without delay and do not cause the oscillations seen with method 1.

However, if one needs to interact with other elements in the scene (collision, manipulation), method 2 would not be able to compensated their effect. The way forward in this case is to modify the simulation engine, which is open source, to give access to all forces and torques before they are applied. This way, the efforts could be modified before physics update, and the compensated effort would be applied. This solution would also avoid the manipulation of individual forces and torques, like it is done in method 2.

Results show that while method 1 is not able to match the dynamic of an UUV, method 2 has body frame velocities similar with the expected *uwv_dynamic_model*.

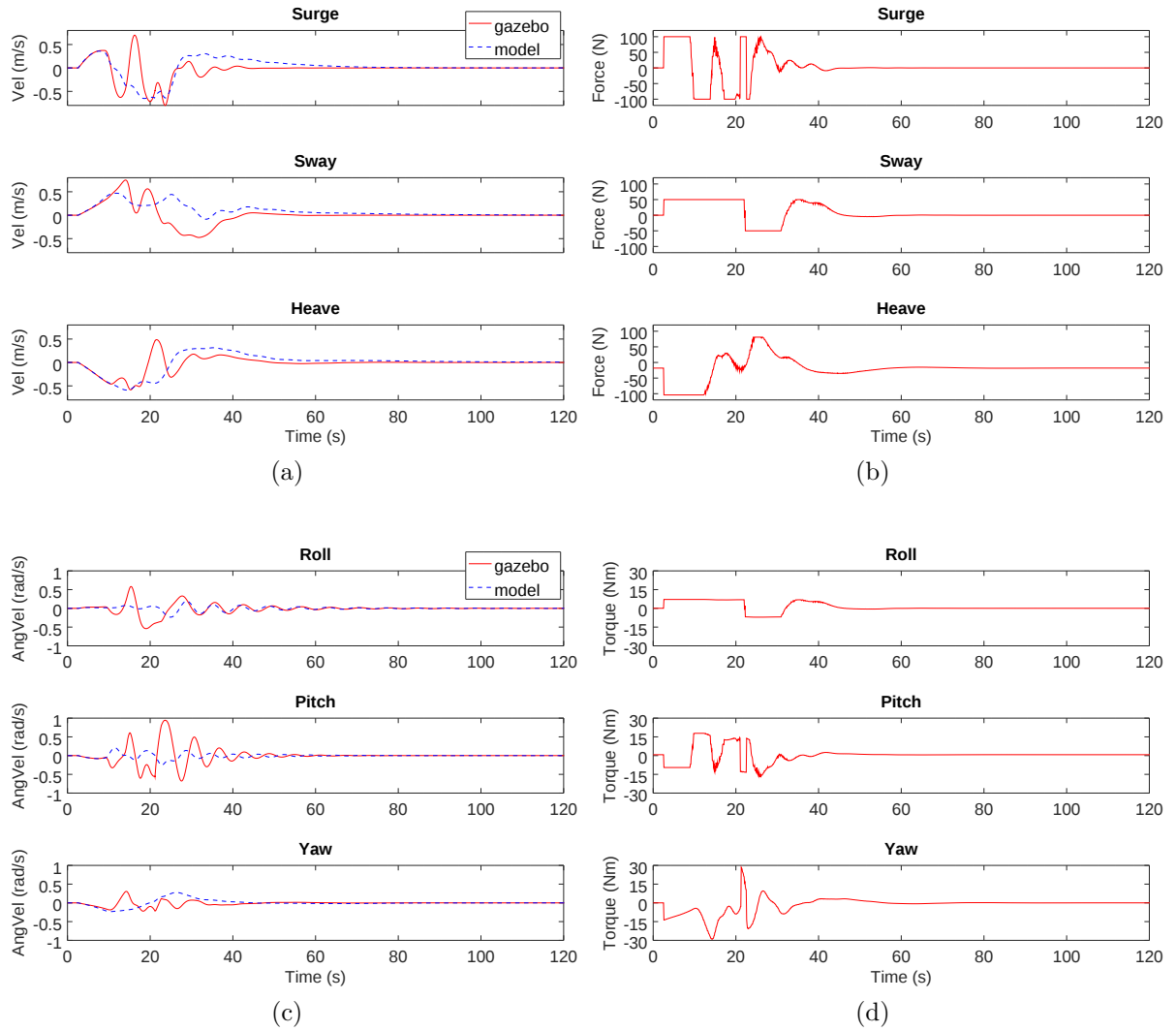


Figure 8 – Method 1: Results of velocities and efforts expressed in body frame: (a) linear velocities, (b) forces, (c) angular velocities, (d) torques

We conclude that it is required to compensate for added mass effects before the simulation step. Delayed compensation has led to a highly oscillatory movement in the simulation.

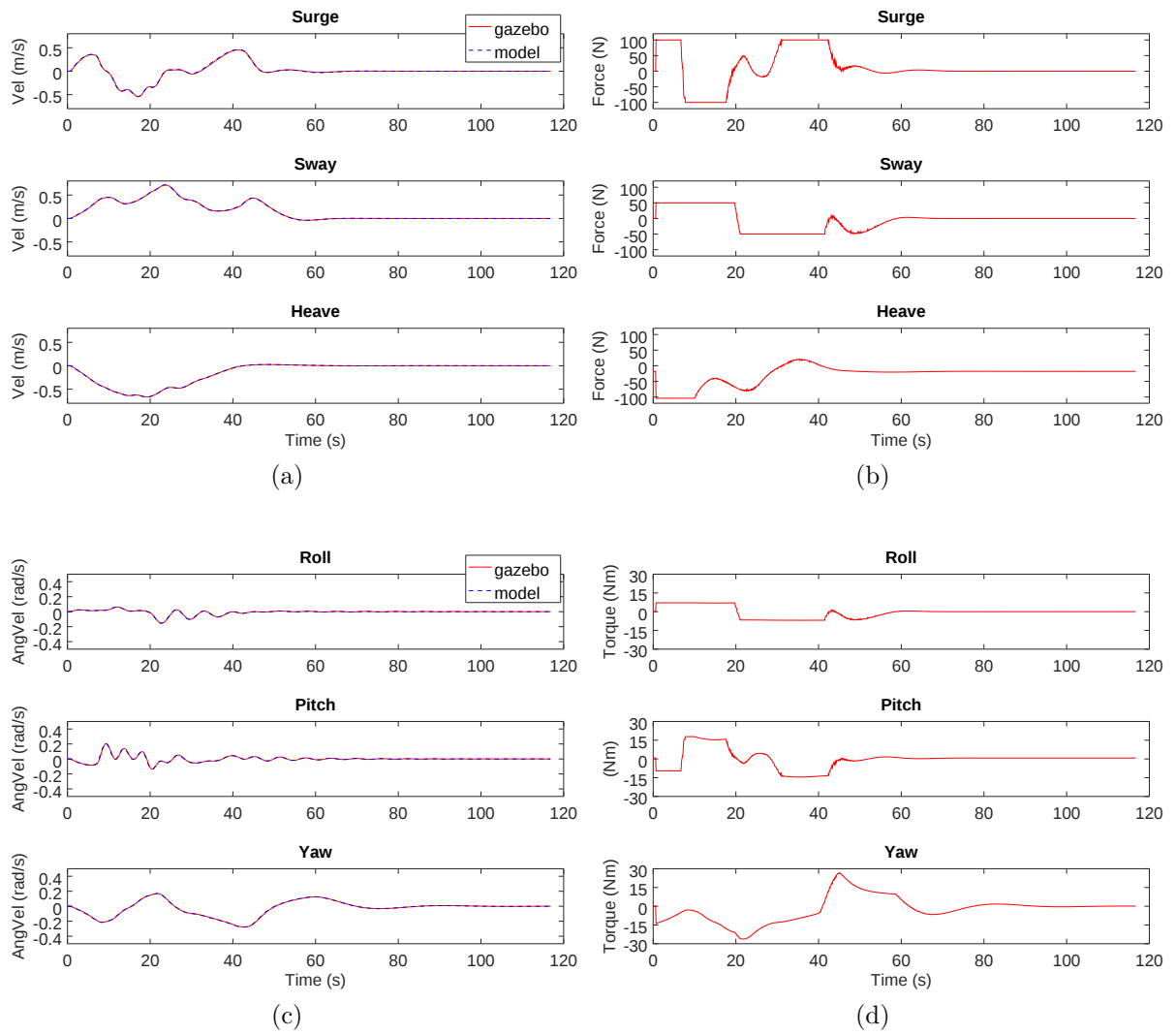


Figure 9 – Method 2: Results of velocities and efforts expressed in body frame: (a) linear velocities, (b) forces, (c) angular velocities, (d) torques

5 Parameters Identification for Underwater Vehicles

This Chapter discuss the problematic of identifying the dynamical model parameters of an underwater vehicle. Section 5.1 is a review of previous works done on parameters identification of UUV and Section 5.2 explores in details the parameters identification methods for the decoupled, single-DOF model represented by Equation 3.12, namely the least square method and the adaptive identifier.

5.1 Literature Review

Determining the parameters of the dynamical model of an underwater vehicle is a complex and laborious activity, due the non-linear characteristic and the coupled terms present in the model. Several techniques of parameters identification has been presented in the literature. The techniques can be summarized in two categories, online or offline, depending on in which moment the identification method is applied, during vehicle's operation or in a post-processing step using logged data. The majority of the works focused on the identification of parameters for the uncouple, single-DOF model. Nevertheless, more recent work concentrate on the parameters identification of the coupled six-DOF model.

The offline parameters identification techniques present in the literature either use Computational Fluid Dynamics (CFD) softwares or are based in experimental logged data.

Software employed in the computation of hydrodynamical terms, like WAMIT [48] and Star-CCM+ [49] were employed in previous work [50], [51]. In [50], WAMIT was used to compute the added mass matrix and Star-CCM+ was used to determine the damping term in a remotely operated vehicle, while [51] also employed WAMIT to compute the added mass matrix of a ROV.

Regarding the offline methods that utilize experimental logged data, it is possible to distinguish between those that use the UUV in a passive way, i.e. where actuation and/or sensing are external to the vehicle, of those who use the UUV in an active way, by employing its embedded propellers and sensors for actuating and collecting data. Among the passive methods, free-decay tests using ropes and/or springs fixed to the UUV are the more common, [52], [53]. Mechanisms for planar movement are also utilized for actuating and collecting data of underwater vehicles [54].

However, the offline methods based on experimental logged data, where the UUV is actuated, appear among the more mentioned in the literature. A common offline parameters identification technique is the least square method, which proven efficacy showed in several

works, [55], [17], [19], [56], [57], [40], including coupled models, [58], [37]. That way, this technique is usually used in evaluation of new parameters identification methods.

Regarding online parameters identification techniques, previous works used neural network and machining learning, Kalman filters, recursive least square and adaptive techniques. In [59], a neural network was used to identify the damping effect of a ROV, [60] also used a neural network to define the dynamic behavior of an UUV, [61] compared several machine learning techniques in order to identify the damping effect of an AUV and [62] used support vector machine (SVM) to identify the parameters of dynamic model.

Among the works that employed Kalman filters, [63] used a dual unscented Kalman filter (DUKF) in a vision guided ROV, [64] applied extended Kalman filter (EKF) in an AUV, [65] used an observer Kalman filter identification (OKID) to identify the parameters of a simplified model of an AUV, and [66] applied three Kalman filter techniques to identify the six DOF model of an AUV.

In [67], a recursive least square method was used for the parameters identification of an AUV.

Among the adaptive parameters identification techniques, [18] used it in a ROV to identify the parameters of a decoupled single-DOF model, [68] employed an adaptive identification method altogether with a fuzzy controller, [69] applied an adaptive parameter identifier to determine the parameters of a rotation rigid body, [20] proposed an adaptive technique to identify the parameters of a coupled 6-DOF model of an UUV, [38] compared adaptive techniques with least square method to identify the damping parameters of an AUV and [40] used an adaptive technique altogether with artificial fidutial markes to identify the model's parameters of an AUV.

Among the several techniques of parameters identification for the dynamical model available, the choose one is the adaptive technique for the uncoupled single-DOF dynamical model presented by [18]. The reasons for that choice are:

1. Convergence proof using Lyapunov-like analysis,
2. It can be applied online,
3. Data of $\dot{v}_i(t)$ is not required, once angular acceleration is usually not instrumented, [70].

Simultaneously, the least square method are applied in order to compare the results.

5.2 1-DOF Model Identification

5.2.1 Least-Squares Parameters Identification

The least square is the classical method of parameters identification, [71]. In this work, it is applied using experimental data of the vehicle's movement in one DOF, requiring data of $\dot{v}_i(t)$, $v_i(t)$ e $u_i(t)$.

Considering $\dot{\bar{V}}_i = [\dot{v}_i(t_1) \cdots \dot{v}_i(t_n)]_{1 \times n}$ e $\bar{F}_i = [f_i(t_1) \cdots f_i(t_n)]_{4 \times n}$ being the data of n samples, Equation 3.14 results in

$$\dot{\bar{V}}_i = \Phi_i^T \bar{F}_i, \quad (5.1)$$

that can be resolved using the pseudo-inverse, case $\bar{F}_i$ is full-rank, by

$$\Phi_i = (\dot{\bar{V}}_i \bar{F}_i^T (\bar{F}_i \bar{F}_i^T)^{-1})^T. \quad (5.2)$$

5.2.2 Adaptive Parameters Identification

In the adaptive parameters identification method presented in [18], given 3.14, an adaptive identifier is given by

$$\dot{\hat{v}}(t) = \hat{\Phi}^T \hat{f}(t) - a_m \Delta v(t), \quad (5.3)$$

where $\hat{f}(t) = [u(t); \hat{v}(t)|\hat{v}(t)|; \hat{v}(t); 1]_{4 \times 1}$, $\hat{v}(t)$ represents the estimation of velocity, $\hat{\Phi}(t)$ represents the lumped parameters Φ and $a_m > 0$ is the gain. The terms representing the coordinated errors are given by

$$\begin{aligned} \Delta v(t) &= \hat{v}(t) - v(t), \\ \Delta \dot{v}(t) &= \dot{\hat{v}}(t) - \dot{v}(t), \\ \Delta \Phi(t) &= \hat{\Phi}(t) - \Phi, \\ \Delta \dot{\Phi}(t) &= \dot{\hat{\Phi}}(t); \quad \dot{\Phi} = 0. \end{aligned} \quad (5.4)$$

The update law is defined as

$$\Delta \dot{\Phi}(t) = -\Lambda \Delta v(t) \hat{f}(t), \quad (5.5)$$

where $\Lambda = \text{diag}[\lambda_i]$, for $i = 1 \cdots 4$, with $\lambda_i > 0$.

The Lyapunov-like lemma, [72], can be used to demonstrate the identifier's stability. The Lyapunov-like lemma states that if a scalar function $V(x, t)$ satisfies the conditions

- $V(x, t)$ is lower bounded,
- $\dot{V}(x, t)$ is negative, and semi-definite

- $\dot{V}(x, t)$ is uniformly continuous in time, i.e, $\ddot{V}(x, t)$ is bounded,

then $\lim_{t \rightarrow \infty} \dot{V}(x, t) = 0$.

Considering the Lyapunov function candidate as

$$V(\Delta v(t), \Delta \Phi(t)) = \Delta v(t)^2 + \Delta \Phi(t)^T \Lambda^{-1} \Delta \Phi(t). \quad (5.6)$$

Using Equations 3.14, 5.3, 5.4, the time derivative of Equation 5.6 is

$$\begin{aligned} \dot{V}(\Delta v(t), \Delta \Phi(t)) &= 2\Delta v(t)\Delta \dot{v}(t) + 2\Delta \Phi(t)^T \Lambda^{-1} \Delta \dot{\Phi}(t) \\ &= 2\Delta v(t)(\dot{\hat{v}}(t) - \dot{v}(t)) + 2\Delta \Phi(t)^T \Lambda^{-1} \Delta \dot{\Phi}(t) \\ &= 2\Delta v(t)(\hat{\Phi}^T(t)\hat{f}(t) - a_m \Delta v(t) - \Phi^T f(t)) + 2\Delta \Phi(t)^T \Lambda^{-1} \Delta \dot{\Phi}(t) \\ &= 2\Delta v(t)((\Delta \Phi^T(t) + \Phi^T)\hat{f}(t) - a_m \Delta v(t) - \Phi^T f(t)) + 2\Delta \Phi(t)^T \Lambda^{-1} \Delta \dot{\Phi}(t) \\ &= -2a_m \Delta v(t)^2 + 2\Delta v(t)\Phi^T(\hat{f}(t) - f(t)) + \\ &\quad 2\Delta v(t)\Delta \Phi^T(t)\hat{f}(t) + 2\Delta \Phi(t)^T \Lambda^{-1} \Delta \dot{\Phi}(t). \end{aligned} \quad (5.7)$$

Using Equation 5.5, then 5.7 becomes

$$\begin{aligned} \dot{V}(\Delta v(t), \Delta \Phi(t)) &= -2a_m \Delta v(t)^2 + 2\Delta v(t)\Phi^T(\hat{f}(t) - f(t)) + \\ &\quad 2\Delta v(t)\Delta \Phi^T(t)\hat{f}(t) - 2\Delta \Phi^T(t)\Lambda^{-1}\Lambda\Delta v(t)\hat{f}(t) \\ &= -2a_m \Delta v(t)^2 + 2\Delta v(t)\Phi^T(\hat{f}(t) - f(t)) + \\ &\quad 2\Delta v(t)\Delta \Phi^T(t)\hat{f}(t) - 2\Delta v(t)\Delta \Phi^T(t)\hat{f}(t) \\ &= -2a_m \Delta v(t)^2 + 2\Delta v(t)\Phi^T(\hat{f}(t) - f(t)) \\ &= -2a_m \Delta v(t)^2 + 2\mu \Delta v(t)^2 + 2\beta \Delta v(t)(\hat{v}(t)|\hat{v}(t)| - v(t)|v(t)|). \end{aligned} \quad (5.8)$$

Examining the term $\Delta v(t)(\hat{v}(t)|\hat{v}(t)| - v(t)|v(t)|)$, it is perceived that when $\hat{v}(t) > v(t)$ implies in $\Delta v(t) > 0$ and $(\hat{v}(t)|\hat{v}(t)| - v(t)|v(t)|) > 0$. When $\hat{v}(t) < v(t)$, it implies in $\Delta v(t) < 0$ and $(\hat{v}(t)|\hat{v}(t)| - v(t)|v(t)|) < 0$. In any case, it results in $\Delta v(t)(\hat{v}(t)|\hat{v}(t)| - v(t)|v(t)|) > 0$.

Since $a_m > 0$, $\mu < 0$ and $\beta < 0$, it implies that $\dot{V}(\Delta v(t), \Delta \Phi(t)) \leq 0$, $V(\Delta v(t), \Delta \Phi(t)) \leq V(\Delta v(0), \Delta \Phi(0))$, and therefore, $\Delta v(t)$ and $\Delta \Phi(t)$ are bounded. Since Φ is constant, $\hat{\Phi}(t)$ is bounded.

As Equation 3.14 is bounded-input bounded-state (BIBS) stable, [18], and given that $u(t)$ is bounded, then $v(t)$ and $\hat{v}(t)$ are also bounded. Consequently $\dot{v}(t)$, $\dot{\hat{v}}(t)$ and $\Delta \dot{v}(t)$ are bounded.

$\ddot{V}(\Delta v(t), \Delta \Phi(t))$ can be determined by making use of $|\dot{x}| = \dot{x} \frac{|x|}{x}$, as

$$\begin{aligned} \ddot{V}(\Delta v(t), \Delta \Phi(t)) &= (-4a_m + 4\mu)\Delta v(t)\dot{v}(t) + 2\beta \Delta \dot{v}(t)(\hat{v}(t)|\hat{v}(t)| - v(t)|v(t)|) + \\ &\quad 2\beta \Delta v(t)(2\dot{\hat{v}}(t)|\hat{v}(t)| - 2\dot{v}(t)|v(t)|). \end{aligned} \quad (5.9)$$

Given that $\ddot{V}(\Delta v(t), \Delta \Phi(t))$ is composed by sums and products of bounded elements, it is therefore bounded.

With the three conditions satisfied, $V \geq 0$, $\dot{V} \leq 0$ and $\ddot{V}$ bounded, the lemma implies that $\lim_{t \rightarrow \infty} \dot{V}(\Delta v(t), \Delta \Phi(t)) = 0$ and consequently $\lim_{t \rightarrow \infty} \Delta v(t) = 0$. As $\hat{f}(t)$ is bounded, it can be concluded from Equation 5.5 that $\lim_{t \rightarrow \infty} \Delta \dot{\Phi}(t) = 0$.

To summarize, all signals stay bounded, the state error, $\Delta v(t)$, and the time-derivative of the parameters error, $\Delta \dot{\Phi}(t)$, converge to zero. Even though the parameters error, $\Delta \Phi(t)$, is bounded, it can not be concluded that it converges to zero as well. However, it is stated in the literature, [72], that the identified parameters can converge to its true values given a sufficiently frequency rich input signal $u(t)$.

6 Experimental Results

This chapter presents the experimental setup, results and discussion of the dynamical model parameters identification using the adaptive identifier and the least-square method, for four DOF: surge, sway, heave and yaw. The experiments were conducted with the two FlatFish prototypes constructed. The first set of experiments was performed in a saltwater test basin using the FlatFish construct in Germany, while the second set of experiments were performed in the sea using the FlatFish vehicle assembled in Brazil.

In sequence, the identified parameters' performance is analyzed considering a multi DOF movement of the vehicle, taking in account three simulated vehicles: a simplified dynamical model, a complete dynamical model and the Gazebo simulation with compensated efforts, presented in Section 4.2.1 as method 2.

6.1 Experiments in Basin

6.1.1 Experimental Setup

The first set of experiments were conducted in the $23m \times 19m \times 8m$ saltwater test basin of the Maritime Exploration Hall at DFKI Robotics Innovation Center (RIC), located in Bremen, Germany [73], on June 10, 2017. The test basin enables an operation without disturbances as water current or waves, allows a good visibility of the vehicle underwater and facilitates the launch and recovery of the AUV through the crane system.

The FlatFish AUV, described in Chapter 2 was used. Two experiments were performed for each controllable DOF: surge, sway, heave and yaw. The first dataset was used for offline parameters identification while the second dataset was intended for cross-validation of the identified parameters. Each experiment consists in a controlled sinusoidal movement, chosen according previous studies [39], [74], [72]. It enables a wide range of velocity for a long period of time while permits the vehicle to move in a safe volume, avoiding possible collision with the bottom and walls of the basin. The sinusoidal parameters were set to generate slow change in the control commands, in order to consider the steady state thruster model.

At the beginning of each experiment, FlatFish was firstly positioned at $\sim 3m$ depth. In sequence, it performed the controlled sinusoidal movement in the DOF of interest while keeping position of the remaining DOFs. The control chain, presented in Section 2.4, was feed with the sinusoidal signal as position set-point. Figure 10 shows FlatFish during one experiment.

The parameters of sinusoidal signals, as the duration for each DOF's experiment,

are presented in Table 5 for the parameters identification dataset and Table 6 for the cross-validation dataset.

Linear velocity data provided by DVL, angular velocity provided by IMU and thruster's rotation speed were recorded. Each sample was timestamped with the measurement time, and once the sensors data were directly used in both identification and cross-validation processes, no delay were perceived or considered.

The Savitz-Golay filter described in Appendix A, was used in order to estimate the vehicle's linear and angular acceleration from the respective velocity data. It was used a filter with 33 samples, being the derivative related to the central sample and using a polynomial of order 3, for all dataset. The computed acceleration was used in the least square method of parameters identification.

With the parameters identified from the first experiment, a numerical simulator, the *uwv_dynamic_simulator* described in Section 4.2.2, was run using those parameters and the logged effort profile of the cross-validation experiment. The error between the velocity predicted by the identified model and the measured velocity during this process is reported as the Mean Absolute Error (MAE), given by

$$MAE = \frac{\sum_{i=1}^n |v_{model_i} - v_{measured_i}|}{n}. \quad (6.1)$$

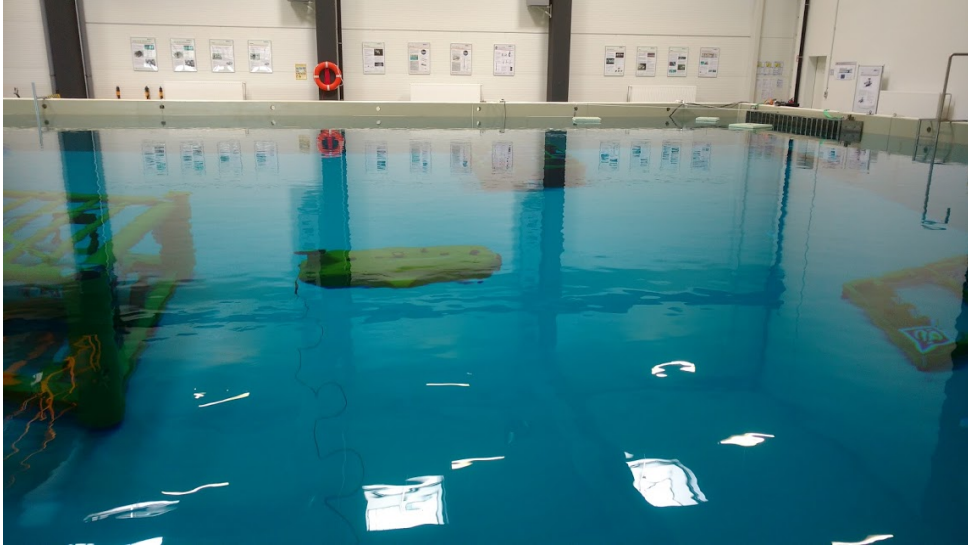


Figure 10 – FlatFish performing experiments in test basin

6.1.2 Results

The adaptive identifier requires initial parameters to be defined at the start of the identification process. In order to explore the capability of the adaptive identifier, it was considered no *a priori* knowledge of the parameters. In such a case, the adaptive identifier was set with initial inertia term equal $100Kg$ and zero for the remaining terms for the

Table 5 – Parameters Identification Experiments

DOF	Duration	Closed Loop Position Control	
		Amplitude	Frequency
Surge	07:10 min	1 m	0.2 rad/s
Sway	05:56 min	1 m	0.2 rad/s
Heave	05:13 min	1 m	0.2 rad/s
Yaw	05:32 min	0.5 rad	0.1 rad/s

Table 6 – Parameters Cross-Validation Experiments

DOF	Duration	Closed Loop Position Control	
		Amplitude	Frequency
Surge	06:49 min	1 m	0.25 rad/s
Sway	05:45 min	1 m	0.25 rad/s
Heave	05:32 min	1 m	0.25 rad/s
Yaw	04:49 min	0.5 rad	0.2 rad/s

four DOF analyzed. The inertia term could not be initialized as zero, once its inverse determines the lumped parameters as shown in Table 3.

The gains used were set empirically and are presented in Table 7. Appendix C explores the influence of gain in the adaptive identifier. An Euler integrator was used in the adaptive identifier and no smooth filter was applied in the output values.

Table 7 – Adaptive Identifier Gains Used in Basin Dataset

DOF	a_m	λ_1	λ_2	λ_3	λ_4
Surge	1	2e-4	5e-1	5e-1	0
Sway	1	2e-4	5e-1	5e-1	0
Heave	1	1.5e-4	5e-1	5e-1	5e-2
Yaw	1	1e-2	1e4	5e0	0

The dataset used in parameters identification (v_i , $\dot{v}_i$ and u_i), the parameters outputted by adaptive identifier and its velocity errors (Δv) are shown in Appendix B.1.

The model's parameters provided by adaptive identifier and least-square methods, as well as the MAE obtained from the cross-validation dataset are presented in Table 8.

Graphs of the cross-validation experiments show the measured velocity altogether with the model's velocity performed with the parameters presents in Table 8, being the Figures 11, 12, 13, 14 representing surge, sway, heave and yaw DOF respectively.

From Table 8, the models whose parameters were identified via adaptive identifier performed slightly better than the ones that used the least-square method for surge, heave and yaw, based on the values of mean absolute error. Both methods identified similar values for the inertia parameters for surge, sway and heave, with a variation of less than

Table 8 – Adaptive and Least-Squares Parameters, Basin Dataset

DOF	Method	Inertia	L. Damping	Q. Damping	Buoyancy	MAE
Surge	Adaptive	763.12	74.77	18.03	0	0.0116 m/s
	LS	797.55	117.45	-381.79	0	0.0123 m/s
Sway	Adaptive	1062.28	134.86	34.05	0	0.0141 m/s
	LS	1035.16	68.14	335.81	0	0.0097 m/s
Heave	Adaptive	1269.13	231.06	39.88	15.59	0.0156 m/s
	LS	1239.34	294.64	-578.74	12.66	0.0176 m/s
Yaw	Adaptive	184.43	2.36	227.92	0	0.0089 rad/s
	LS	233.26	3.72	167.40	0	0.0164 rad/s

5%. The FlatFish has about 275kg of dry mass. It mean that FlatFish has an added mass of about 500kg in surge, 775kg in sway and 975kg in heave. Those values seem to be reasonable since the cross section area of surge, sway and heave are respectively $\sim 0.525m^2$, $\sim 1.1m^2$ and $\sim 2.31m^2$. As the added mass is related to the water that is accelerated by the AUV while it accelerates, the smaller the cross section area, the smaller the added mass.

The damping terms seems to have a combined effect. The two damping parameters provided by both methods could not be individually compared, since a damping term has different values depending on the identification method used for the same DOF. For the surge and heave DOF, the least-square method provided negative parameters for the quadratic damping, which does not make physical sense once it implies that this individual damping term would apply energy to the system.

In Figure 25 one can see at second 70 for example, what is believed to be the reason for the negative damping. At that moment, the force is near zero, about to flip from positive to negative value while the acceleration has already a negative value, meaning that the thrusters was still applying a positive force when the vehicle starts its deceleration. For the least square method, the damping term was adding a negative force and causing vehicle's deceleration despite the positive force provided by thruster. The reason for this timing difference is not clear, since timestamped measured data was direct used for velocity and to derive acceleration, as well to determine force from thurster's rotation speed. It is believed that either unmodeled thruster's dynamics or ignored vehicle's dynamic may be the cause for the timing offset between force and acceleration.

Nevertheless, due the bounded region of operation, the combination of both damping parameters has an equivalent result for both identification methods, based on the graphics and MAE resultant of the cross-validation experiment.

A graphical overview of Figures 11, 12, 13 shows that for surge, sway and heave DOF, both parameters identifiers have similar results. The yaw DOF in Figure 14 performed worst than the others DOFs. It is believed that the small amplitude of torque and velocity

in the identification dataset show in Figure 31, of $\sim 1.5Nm$ and $\sim 0.05rad/s$ respectively, was not enough to fully capture the dynamic effects of an operation with amplitudes of $\sim 5Nm$ for torque and $\sim 0.1rad/s$ for velocity observed in Figure 14.

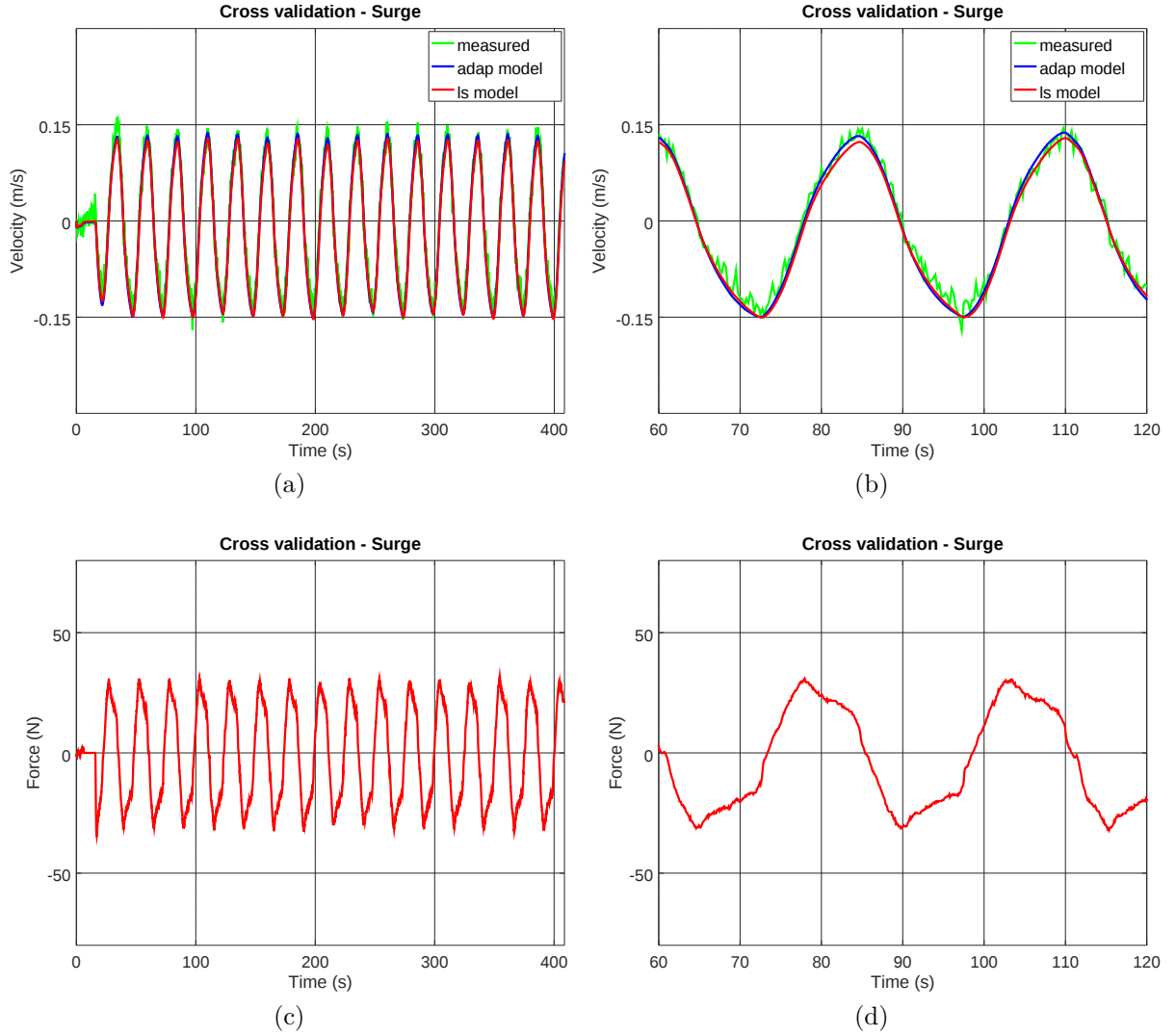


Figure 11 – Cross validation in surge performed in basin: (a) Velocity, full experiment, (b) Velocity, 60s magnification, (c) Applied force, full experiment, (d) Applied force, 60s magnification

6.2 Experiments in Bay

6.2.1 Experimental Setup

The second set of experiments were conducted in Marina de Aratu, located in Aratu bay, Bahia, Brazil, on December 5, 2017. Figure 15 shows in a map the location where the experiments were performed. The test site has relatively low perturbations, such as waves and sea currents, and an average depth of $5m$, [75]. The experiments were conducted from Lady Catarina, a catamaran equipped with laboratory infrastructure that allowed

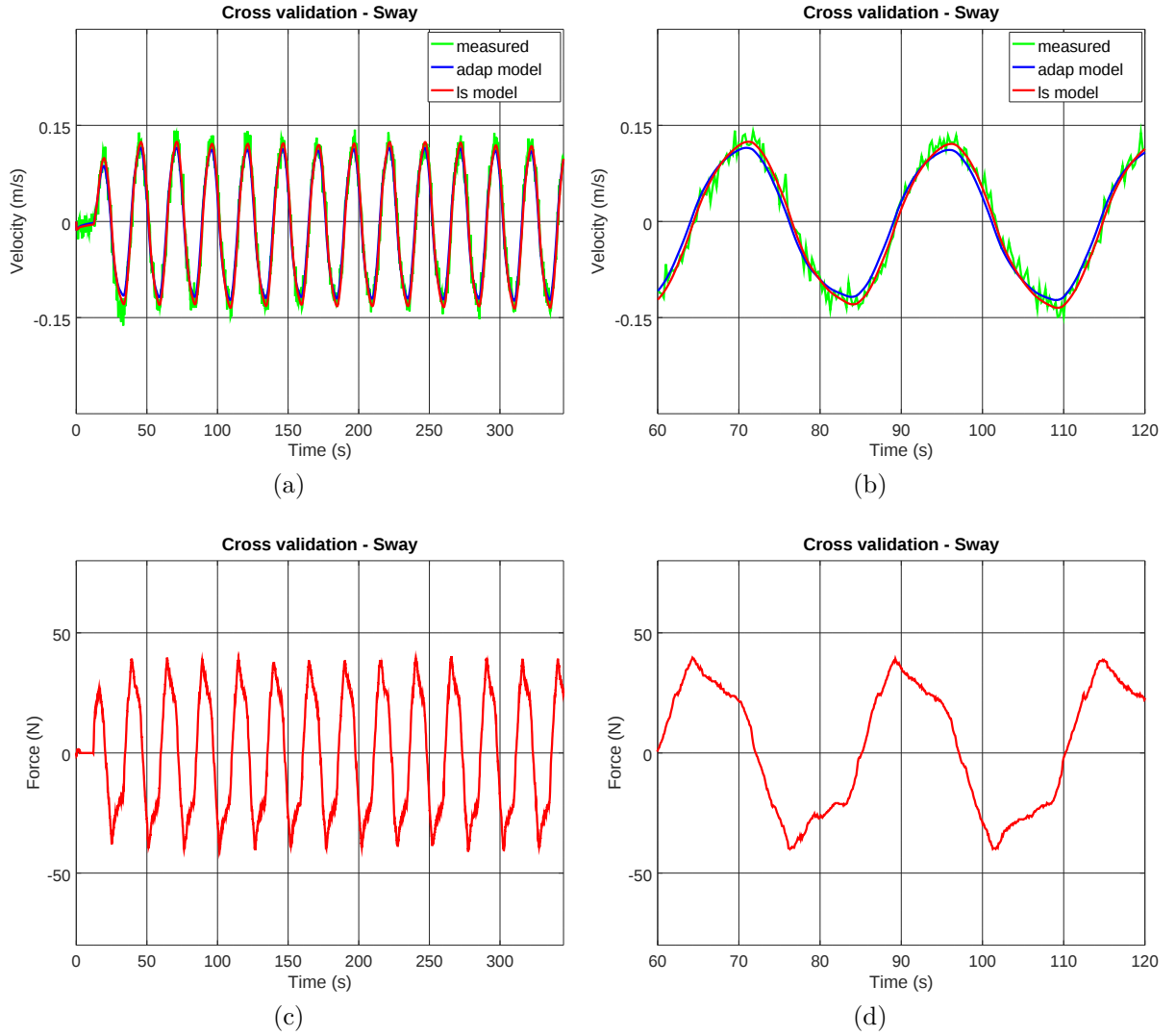


Figure 12 – Cross validation in sway performed in basin: (a) Velocity, full experiment, (b) Velocity, 60s magnification, (c) Applied force, full experiment, (d) Applied force, 60s magnification

FlatFish's operation. The vessel was docked in the Marina with a reserved area for FlatFish, as shown in Figure 16. Figure 17 shows FlatFish being prepared for experiments.

The experimental setup described in Section 6.1.1 was applied for identification and cross-validation experiments in the bay location as well, with different sinusoidal parameters. The parameters of input signals, as the duration for each DOF's experiment, are presented in Table 9 for the parameters identification dataset and Table 10 for the cross-validation dataset.

The sinusoidal parameters were set with the purpose of generating slow change in the control commands, in order to consider the steady state thruster model. Nevertheless, the vehicle's control chain performed improperly causing abrupt and punctual changes in the control commands, as can be seen in Figure 18 for example.

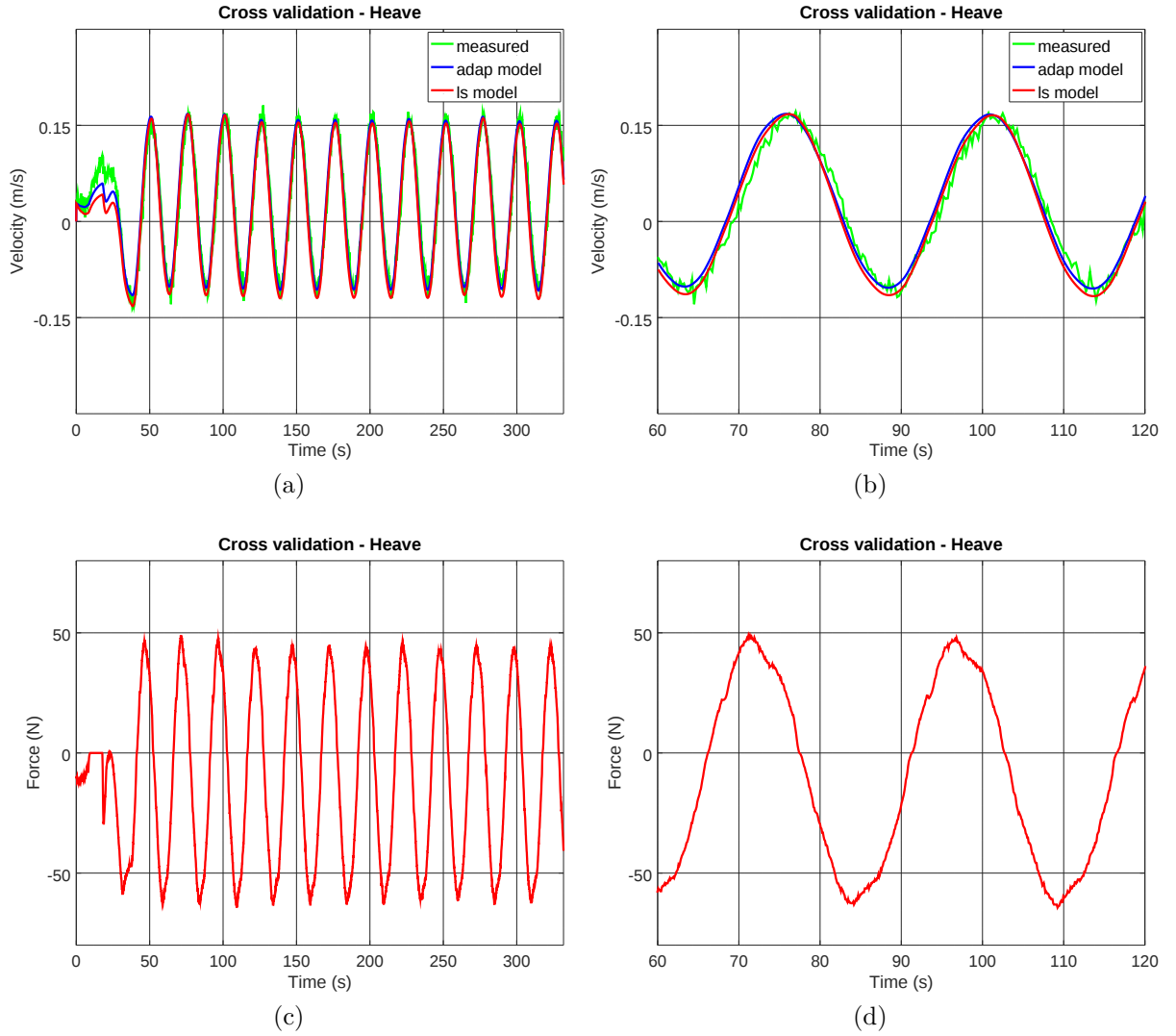


Figure 13 – Cross validation in heave performed in basin: (a) Velocity, full experiment, (b) Velocity, 60s magnification, (c) Applied force, full experiment, (d) Applied force, 60s magnification

6.2.2 Results

The adaptive identifier requires initial parameters to be defined at the start of the identification process. In order to explore the capability of the adaptive identifier, it was considered no *a priori* knowledge of the parameters. In such a case, the adaptive identifier was set with initial inertia term equal $100Kg$ and zero for the remaining terms for the four DOF analyzed. The inertia term could not be initialized as zero, once its inverse determines the lumped parameters as shown in Table 3.

The gains used were set empirically and are presented in Table 11. Appendix C explores the influence of gain in the adaptive identifier. An Euler integrator was used in the adaptive identifier and no smooth filter was applied in the output values.

The dataset used in parameters identification (v_i , $\dot{v}_i$ and u_i), the parameters

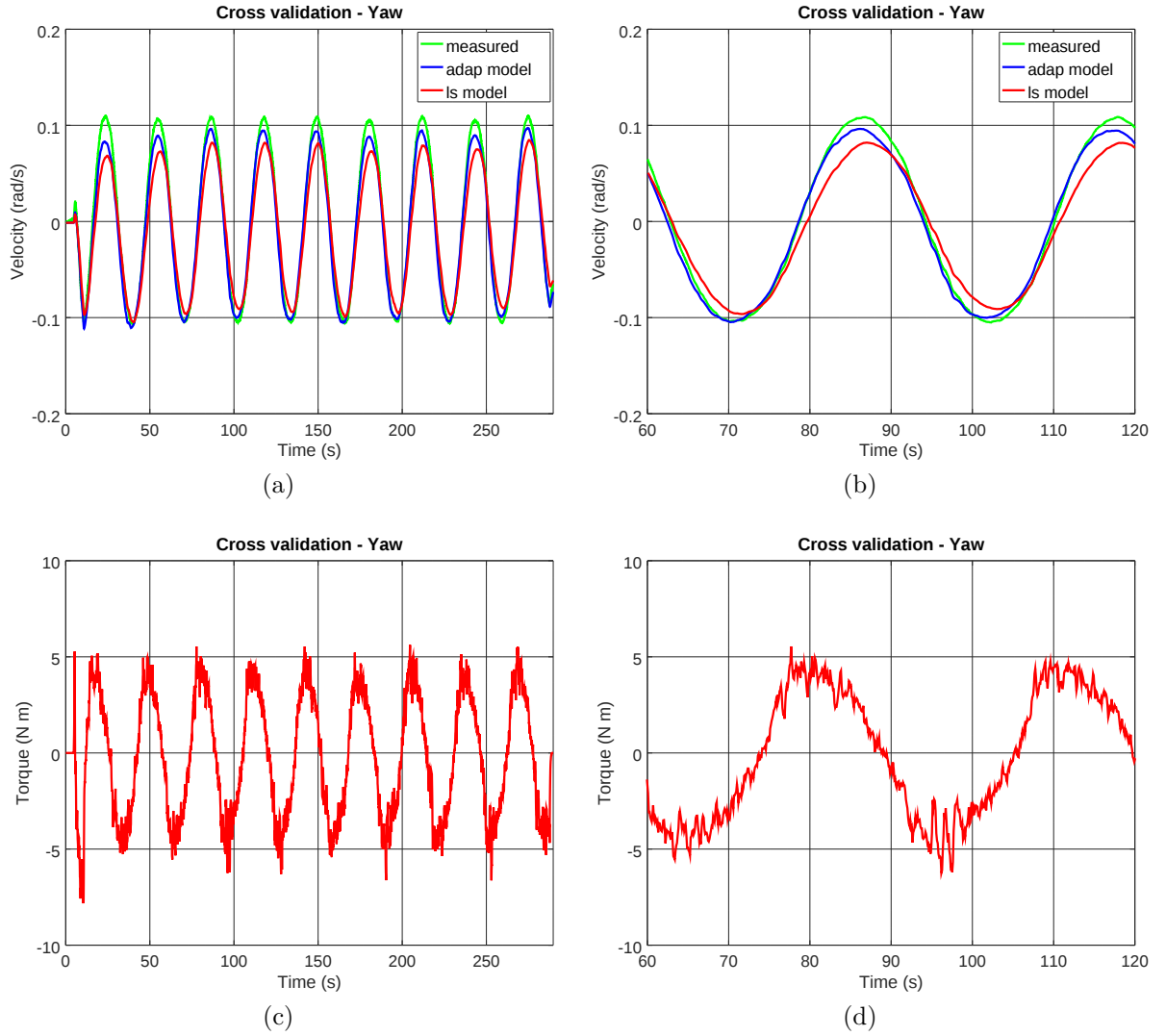


Figure 14 – Cross validation in yaw performed in basin: (a) Velocity, full experiment, (b) Velocity, 60s magnification, (c) Applied torque, full experiment, (d) Applied torque, 60s magnification

provided by adaptive identifier and its velocity errors (Δv) are shown in Appendix B.2.

The model's parameters provided by adaptive identifier and least-square methods, as well as the MAE obtained from the cross-validation dataset are presented in Table 12.

Graphs of the cross-validation experiments show the measured velocity altogether with the model's velocity performed with the parameters presents in Table 12, being the Figures 18, 19, 20, 21 representing surge, sway, heave and yaw DOF respectively.

From Table 12, the models whose parameters were identified via adaptive identifier performed slightly better than the ones that used the least-square method for sway and yaw, based on the values of mean absolute error. Both methods identified similar values for the inertia parameters for sway, heave and yaw. The FlatFish has about $275kg$ of dry mass. It mean that FlatFish has an added mass of about $525kg$ in surge, $795kg$ in sway

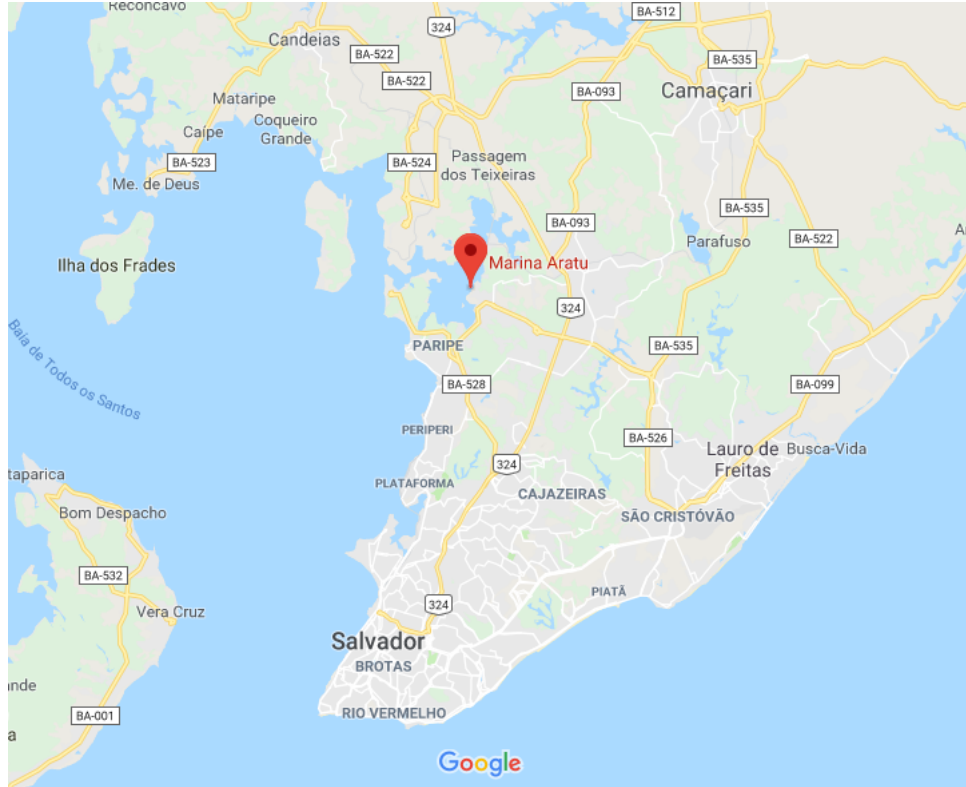


Figure 15 – Test site located at Marina de Aratu in Aratu bay (Google Maps)

Table 9 – Parameters Identification Experiments

DOF	Duration	Closed Loop Position Control	
		Amplitude	Frequency
Surge	04:57 min	2 m	0.2 rad/s
Sway	04:45 min	2 m	0.2 rad/s
Heave	04:44 min	1 m	0.15 rad/s
Yaw	04:59 min	1 rad	0.2 rad/s

Table 10 – Parameters Cross-Validation Experiments

DOF	Duration	Closed Loop Position Control	
		Amplitude	Frequency
Surge	04:44 min	2 m	0.15 rad/s
Sway	05:07 min	2 m	0.15 rad/s
Heave	04:56 min	1 m	0.13 rad/s
Yaw	05:06 min	1 rad	0.15 rad/s

and $1075kg$ in heave. Those values of inertia are equivalents with the inertia parameters found for the German vehicle, shown in Table 8.

The damping terms seems to have a combined effect. In general, the two damping parameters provided by both methods could not be individually compared, with an exception of the linear damping in surge. For sway, the least-square method provided a negative parameter for the quadratic damping, which does not make physical sense once it



Figure 16 – Catamaran used as laboratory, docked at Marina de Aratu



Figure 17 – FlatFish preparing for experiments in bay

implies that this individual damping term would apply energy to the system.

In Figure 35 one can see at second 70 for example, what is believed to be the reason for the negative damping. At that moment, the force is near zero, about to flip from positive to negative value while the acceleration has already a negative value, meaning that the thrusters was still applying a positive force when the vehicle starts its deceleration. For the least square method, the damping term was adding a negative force and causing vehicle's deceleration despite the positive force provided by thruster. The reason for this timing difference is not clear, since timestamped measured data was direct used for velocity

Table 11 – Adaptive Identifier Gains Used in Bay Dataset

DOF	a_m	λ_1	λ_2	λ_3	λ_4
Surge	1	2e-4	5e-1	5e-1	0
Sway	1	2e-4	5e-1	5e-1	0
Heave	1	1.5e-4	5e-1	5e-1	5e-2
Yaw	1	5e-4	5e1	5e-1	0

Table 12 – Adaptive and Least-Squares Parameters, Bay Dataset

DOF	Method	Inertia	L. Damping	Q. Damping	Buoyancy	MAE
Surge	Adaptive	758.66	94.32	27.27	0	0.0205 m/s
	LS	854.38	93.32	53.86	0	0.0163 m/s
Sway	Adaptive	1081.31	147.11	51.50	0	0.0225 m/s
	LS	1061.07	189.15	-445.32	0	0.0316 m/s
Heave	Adaptive	1381.98	218.75	39.88	33.98	0.0201 m/s
	LS	1301.02	169.02	235.29	30.59	0.0094 m/s
Yaw	Adaptive	304.27	11.24	236.12	0	0.0225 rad/s
	LS	312.52	27.60	152.43	0	0.0316 rad/s

and to derive acceleration, as well to determine force from thruster's rotation speed. It is believed that either unmodeled thruster's dynamics or ignored vehicle's dynamic may be the cause for the timing offset between force and acceleration.

Nevertheless, due the bounded region of operation, the combination of both damping parameters has an equivalent result for both identification methods.

The buoyancy of the Brazilian vehicle is about $30N$, twice that of the German FlatFish with buoyancy of $\sim 15N$. It is reasonable that each vehicle has its own value of buoyancy due a fine trim performed for each test location.

A graphical overview of Figures 18, 20, 21 shows that for surge, heave and yaw, both parameters identifiers have similar results. The sway DOF in Figure 19 performed worst than the others DOFs. It is believed that a current due to tide change may be responsible for affecting the expected velocity provided by the identified model's parameters.

The abrupt and punctual changes in the control commands caused by malfunctions in the control chain were not a problem for both the identification and the cross-validation procedure. Instead, they have shown to enrich the analyses, specifically for the cross-validation dataset. In those cases, it is observed that the models using the identified parameters behave according the measured data for those punctual and abrupt changes.

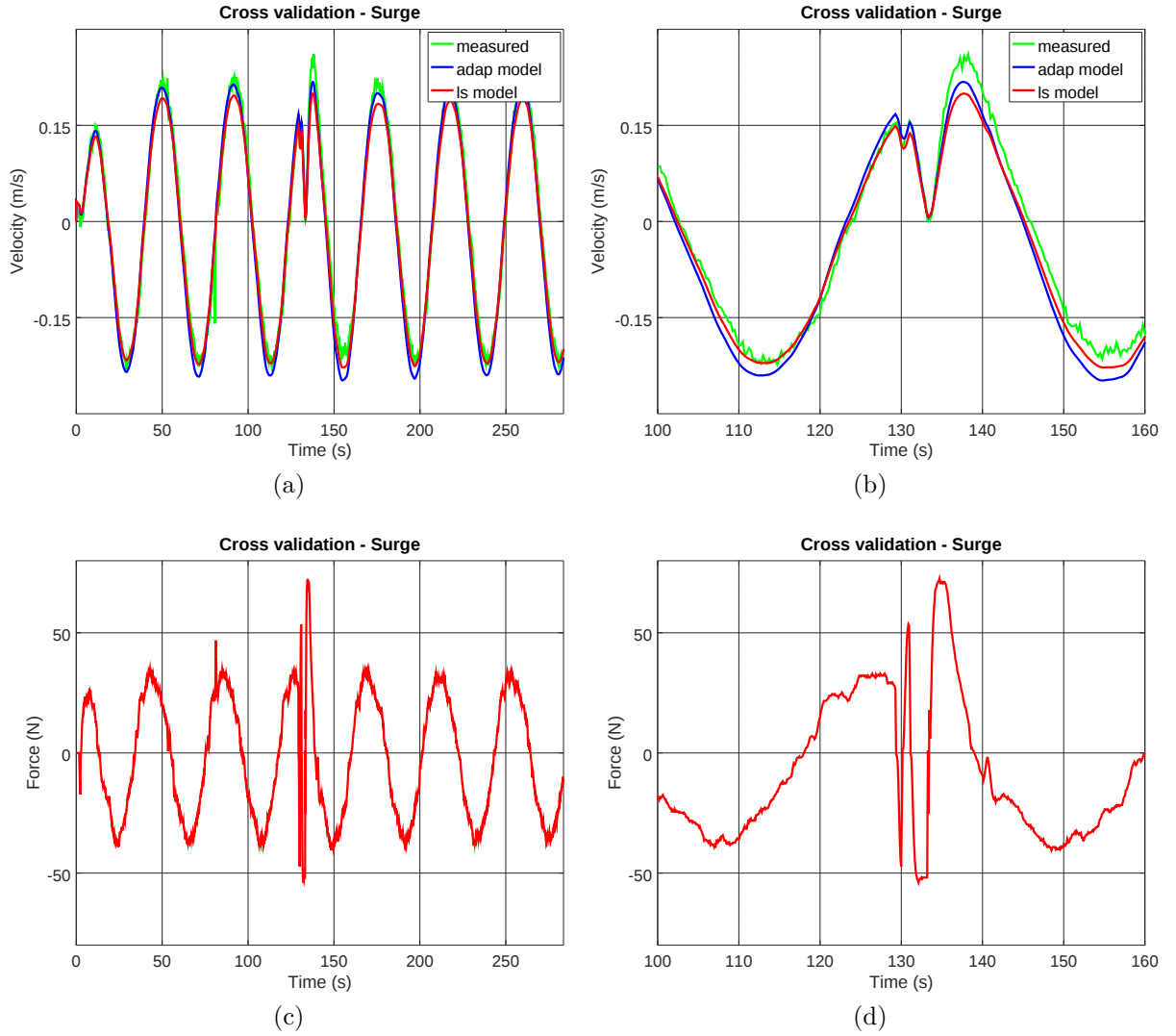


Figure 18 – Cross validation in surge performed in bay: (a) Velocity, full experiment, (b) Velocity, 60s magnification, (c) Applied force, full experiment, (d) Applied force, 60s magnification

6.3 Multi DOF Actuation

With the parameters of the simplified dynamic model identified for four DOF of FlatFish and the shown performance of the given model when the vehicle actuates in one DOF, the question of how this model's parameters behaves in a multi DOF movement of the vehicle arises. This section explores how the simplified dynamic model and the complete dynamic model, both configured with the same identified parameters, behaves compared with measured velocities in a multi DOF movement of FlatFish. The complete dynamic model is played both by a dedicated library for UUV (*uwv_dynamic_model*) and by Gazebo simulation adapted to consider the added mass effect.

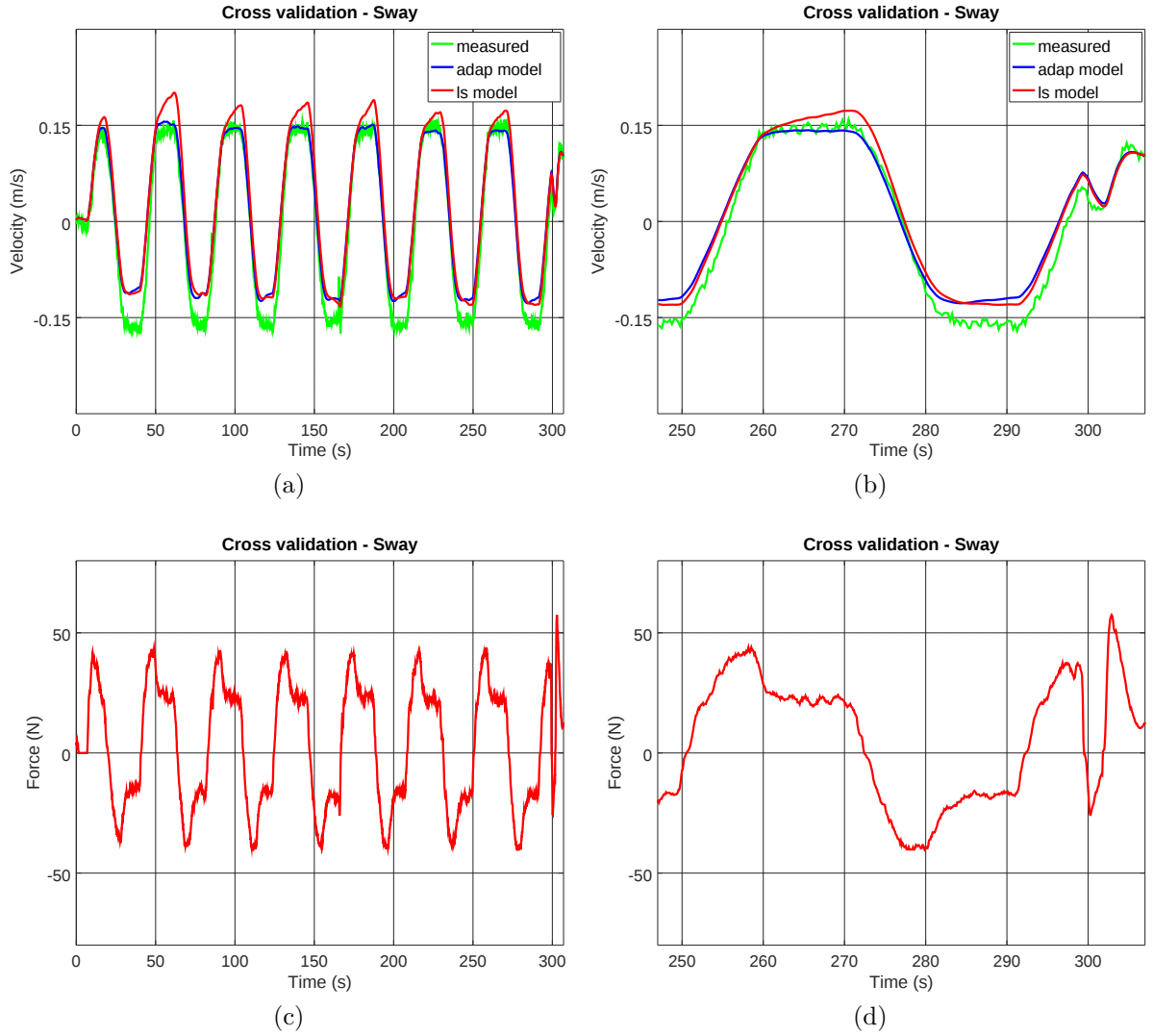


Figure 19 – Cross validation in sway performed in bay: (a) Velocity, full experiment, (b) Velocity, 60s magnification, (c) Applied force, full experiment, (d) Applied force, 60s magnification

6.3.1 Experimental Setup

The experiments were conducted in the same saltwater test basin of DFKI used for model's parameters identification, on June 10, 2017.

The same FlatFish AUV used in the identification procedure in basin was used. One experiment was performed where all the four controllable DOF were actuated at the same time, in controlled sinusoidal movements. It enables a wide range of velocity for a long period of time while permits the vehicle to move in a safe volume, avoiding possible collision with the bottom and walls of the basin. The sinusoidal parameters were set to generate slow change in the control commands, in order to consider the steady state thruster model.

At the beginning of each experiment, FlatFish was firstly positioned at $\sim 3\text{m}$ depth.

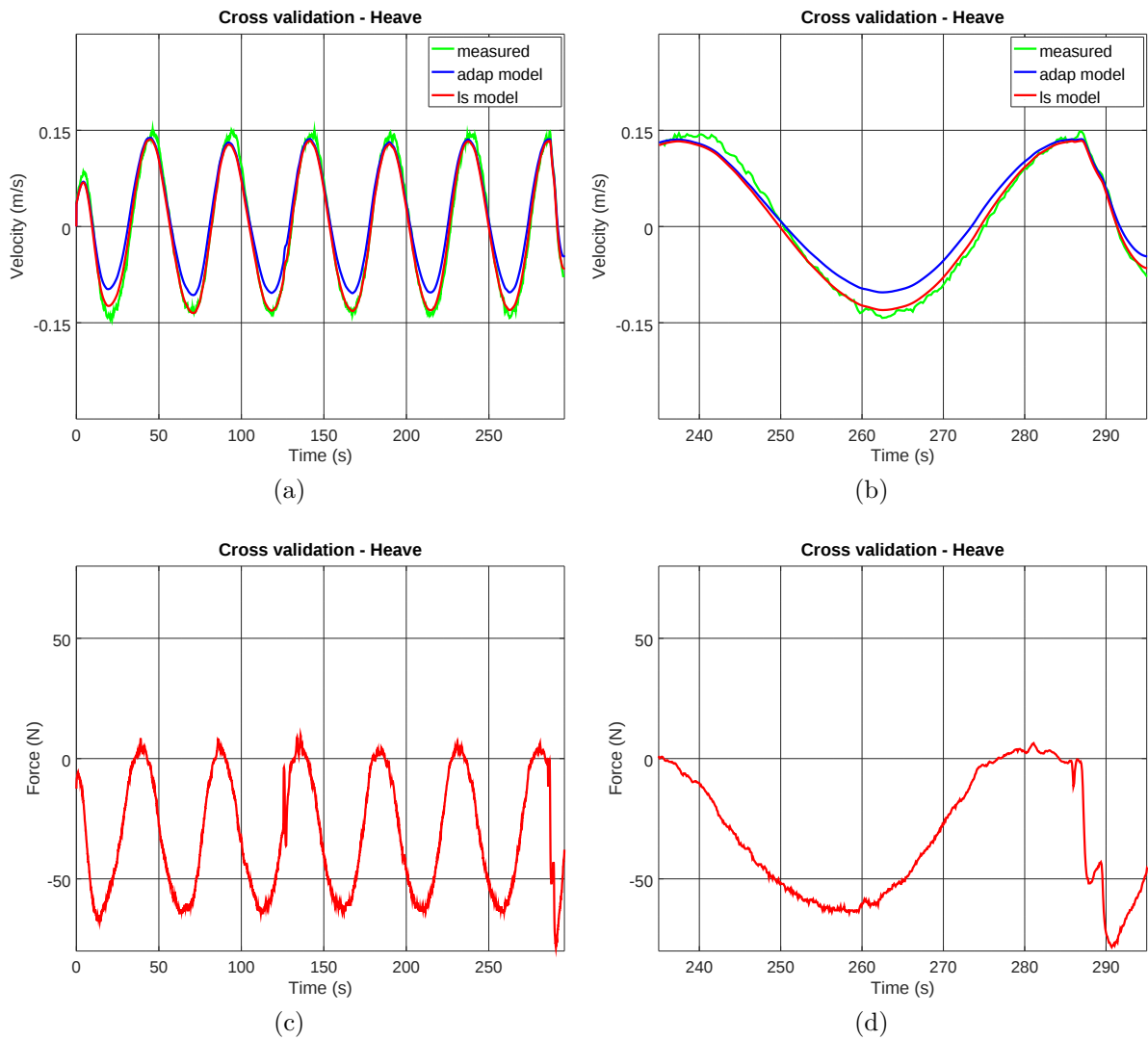


Figure 20 – Cross validation in heave performed in bay: (a) Velocity, full experiment, (b) Velocity, 60s magnification, (c) Applied force, full experiment, (d) Applied force, 60s magnification

In sequence, it performed the controlled sinusoidal movement in four DOF. The control chain, presented in Section 2.4, was feed with four sinusoidal signal as position set-point, one for each DOF.

The parameters of sinusoidal signals, as the duration of the experiment, are presented in Table 13.

Table 13 – Multi DOF Experiment

DOF	Duration	Closed Loop Position Control Amplitude	Frequency
Surge	05:42 min	1 m	0.3 rad/s
Sway		1 m	0.25 rad/s
Heave		1 m	0.2 rad/s
Yaw		0.5 rad	0.2 rad/s

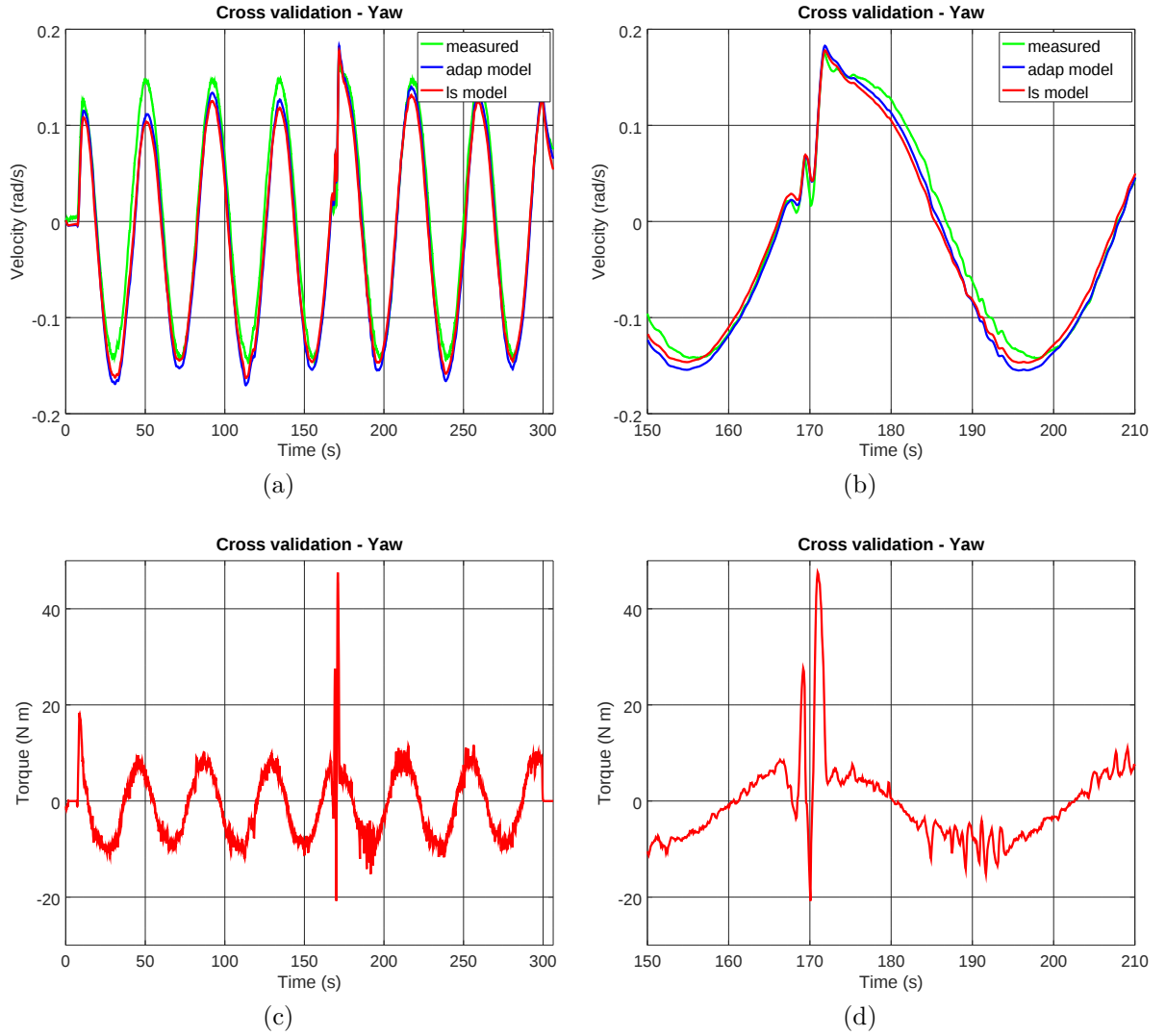


Figure 21 – Cross validation in yaw performed in bay: (a) Velocity, full experiment, (b) Velocity, 60s magnification, (c) Applied torque, full experiment, (d) Applied torque, 60s magnification

Linear velocity data provided by DVL, angular velocity provided by IMU and thruster's rotation speed were recorded. Each sample was timestamped with the measurement time, and once the sensors data were directly used, no delay were perceived or considered.

The logged efforts profile was used to feed three numerical simulators:

- Simplified model, given by Equation 3.12,
- Complete model, given by Equation 3.10,
- Gazebo simulation with compensated efforts, given by Equations 4.1 and 4.15.

The model's parameters identified via the adaptive identifier, provided in Table 8, were used. For roll and pitch DOF, whose identification procedure could not be performed,

their parameters were arbitrarily defined. The inertia matrix, linear and quadratic damping, used in the three simulations, are described respectively by

$$M = \begin{bmatrix} 763.12 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1062.28 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1269.13 & 0 & 0 & 0 \\ 0 & 0 & 0 & 200 & 0 & 0 \\ 0 & 0 & 0 & 0 & 200 & 0 \\ 0 & 0 & 0 & 0 & 0 & 184.43 \end{bmatrix}, \quad (6.2)$$

$$D_L = \begin{bmatrix} 74.77 & 0 & 0 & 0 & 0 & 0 \\ 0 & 134.86 & 0 & 0 & 0 & 0 \\ 0 & 0 & 231.06 & 0 & 0 & 0 \\ 0 & 0 & 0 & 200 & 0 & 0 \\ 0 & 0 & 0 & 0 & 200 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2.36 \end{bmatrix}, \quad (6.3)$$

$$D_Q = \begin{bmatrix} 18.03 & 0 & 0 & 0 & 0 & 0 \\ 0 & 34.05 & 0 & 0 & 0 & 0 \\ 0 & 0 & 39.88 & 0 & 0 & 0 \\ 0 & 0 & 0 & 200 & 0 & 0 \\ 0 & 0 & 0 & 0 & 200 & 0 \\ 0 & 0 & 0 & 0 & 0 & 227.92 \end{bmatrix}. \quad (6.4)$$

The resultant buoyancy is 15.59N.

6.3.2 Results

Figures 22 and 23 show body frame velocities and efforts, for linear and angular DOF, respectively, of FlatFish's measured data during experiment, altogether with simple model, complete model and Gazebo simulation. Table 14 shows the mean absolute error, given by Equation 6.1, for all six DOF of the three simulated models.

It is perceived that for surge, sway and yaw DOF, the behavior of all models' velocities mismatch vehicle's measurement, being those DOF with the higher MAE in Table 14. Heave is the DOF with more similarity between model's velocities and measurements, while roll and pitch DOF shows small amplitudes of movement but still with discrepancy between measurement and model's velocities.

Heave DOF does not show to be coupled with the others three controllable DOFs. As the vehicle performed with small amplitudes in roll and pitch, velocities in heave of models were evenness with measurement. It is believed that the highly coupled movement in surge, sway and yaw contributes to the model's inability to perform with a similar behavior to the real vehicle in these DOFs.

Table 14 – Mean Absolute Error in Multi DOF Experiment

DOF	Simple model	Complete model	Gazebo
Surge	0.0391 m/s	0.0498 m/s	0.0496 m/s
Sway	0.0264 m/s	0.0246 m/s	0.0245 m/s
Heave	0.0198 m/s	0.0200 m/s	0.0207 m/s
Roll	0.0126 rad/s	0.0124 rad/s	0.0125 rad/s
Pitch	0.0124 rad/s	0.0110 rad/s	0.0112 rad/s
Yaw	0.0489 rad/s	0.0492 rad/s	0.0493 rad/s

For the simplified model, where the Coriolis and centripetal term is ignored, it is clear to understanding the reason of such mismatch. Nevertheless, for the complete model, where the Coriolis terms are considered, it can be concluded that the identified parameters for a simplified model does not consider all the necessary dynamics for such coupled movement. It raises the question if parameters provided by a complete model's parameters identification method would result in a more reliable simulation for the multi DOF case. Still, it is important to remark in Figures 22 and 23 the akin behavior of the Gazebo simulation with compensated efforts compared to the complete model. It indicates the ability of Gazebo simulation to reproduce the dynamics of a real UUV in case the complete model's parameters were correctly identified, which can be objective of a future work.

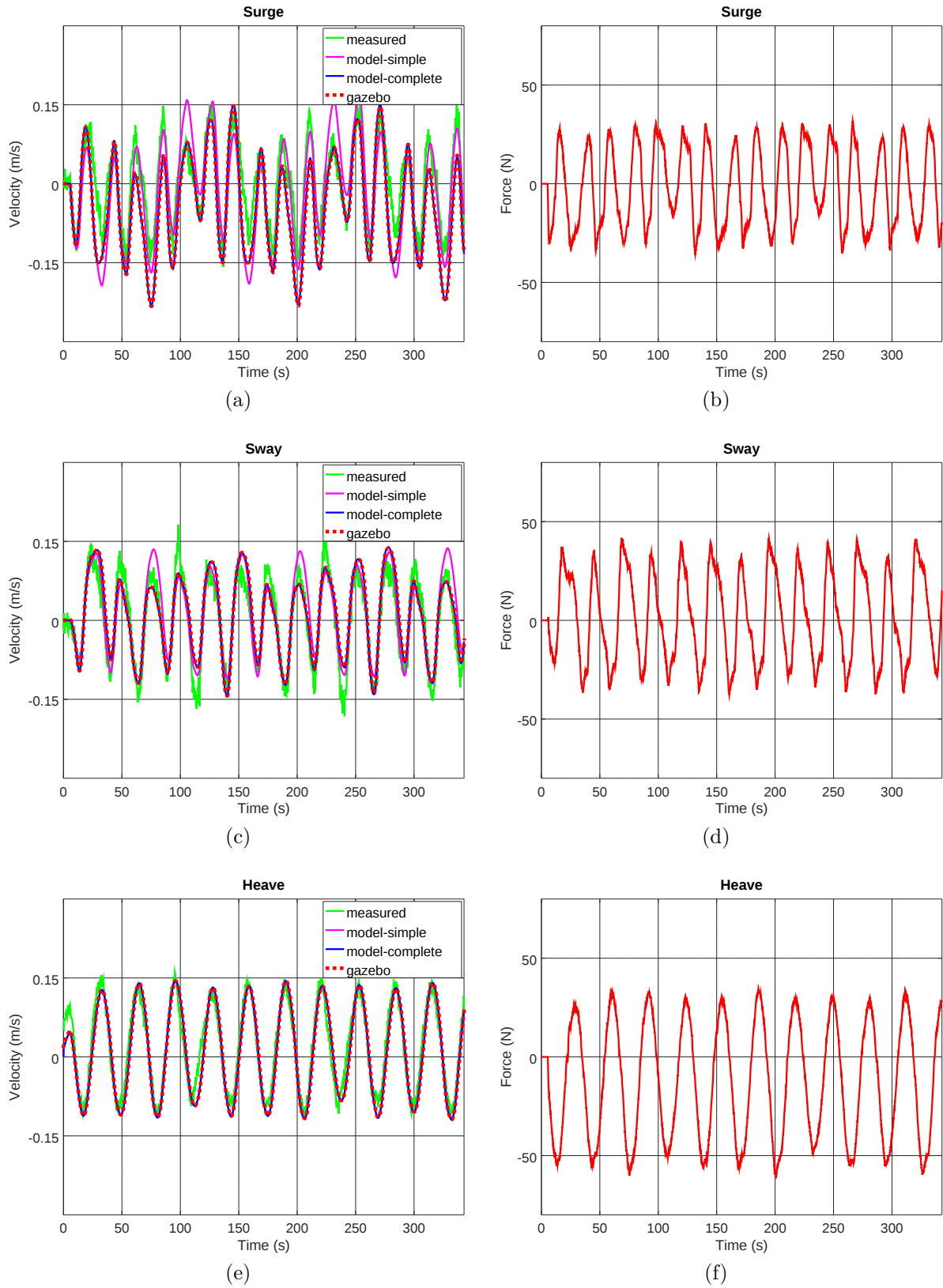


Figure 22 – Multi DOF actuation in basin, linear DOF: (a) Velocity surge, (b) Force surge, (c) Velocity sway, (d) Force sway, (e) Velocity heave, (f) Force heave

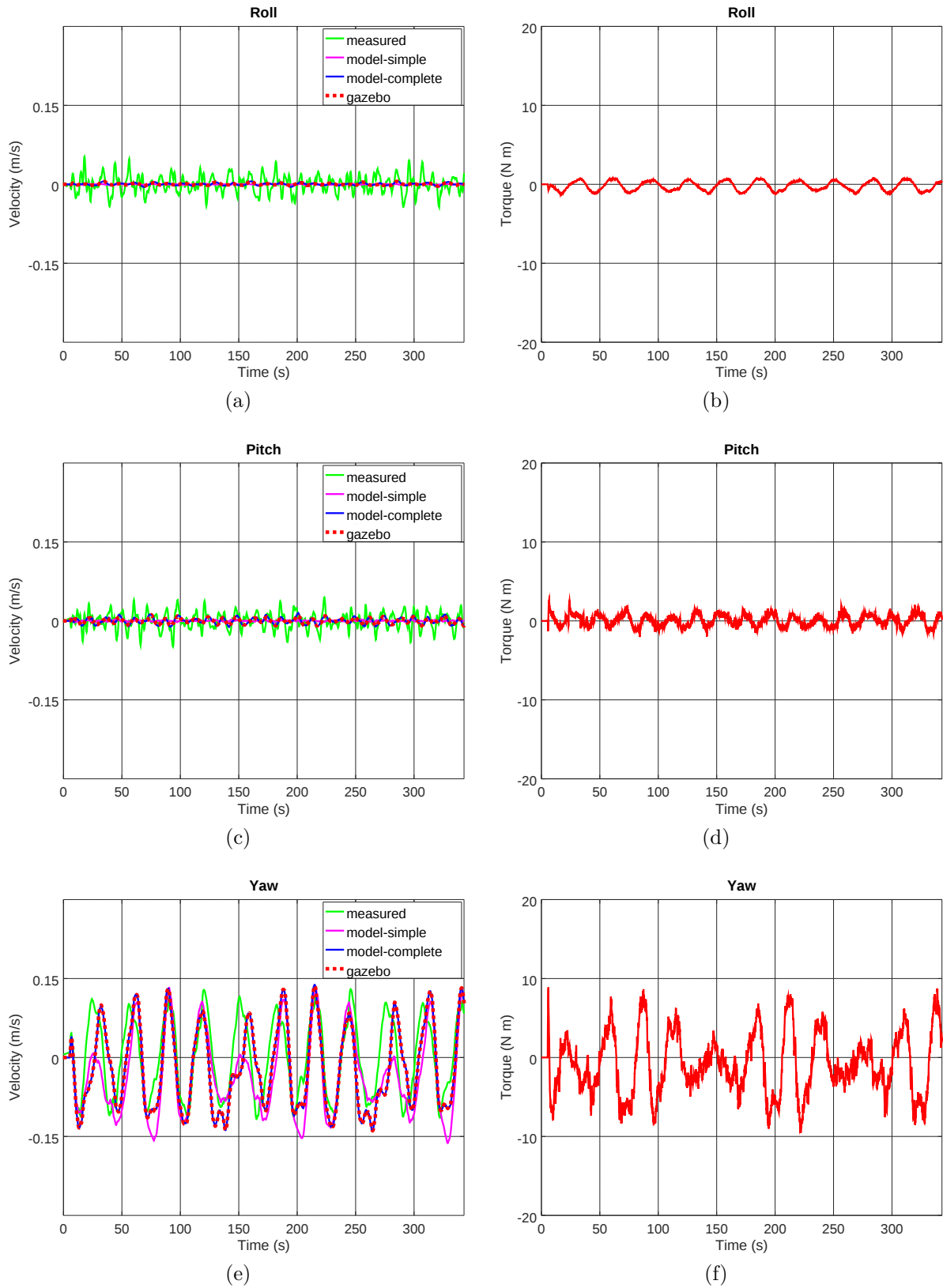


Figure 23 – Multi DOF actuation in basin, angular DOF: (a) Angular velocity roll, (b) Torque roll, (c) Angular velocity pitch, (d) Torque pitch, (e) Angular velocity yaw, (f) Torque yaw

7 Conclusion

This work presented important results regarding parameters identification and simulation of an AUV. In introduction, the importance of simulation environment during the development of UUV is explored, altogether with the necessity of identifying the parameters of the dynamic model, not only to improve the simulation but also to enhance model-based controllers, state estimator and model-based fault detection. In the sequence, a description of the FlatFish AUV is presented, in which the sensors, actuators, software and controller employed were presented.

The dynamical model of an underwater vehicle were shown, in which the main hydrodynamic terms, such as added mass, damping and gravitational effects were presented. Considerations and simplifications were made in order to obtain a decoupled single DOF dynamical model.

The simulation of underwater robots were also explored. It was perceived a representation lack of the added mass effect in the main robotic simulators in existence. The added mass, although being considered in dedicated marine simulators, could not be represented in widespread robotic simulators such Gazebo, due the limitations of the physic engine. A method to overcome that constrain was presented and have already been published in a paper [76].

In the following, techniques for parameters identification used in underwater vehicles were studied. The least square method and the adaptive identifier were formulated for the decoupled single DOF model. A stability analysis of the adaptive identifier was defined.

Regarding experimental results, the techniques for parameters identification were applied in two FlatFish vehicles, with data collected from experiments performed in two different location, a saltwater basin located in Bremen, Germany, and Aratu bay, nearby Salvador, Brazil. Both techniques provided similar results in the cross-validation experiments. The experiments performed in Aratu bay was subjected to tide movement, causing the performance of identification techniques to be not as good as the ones performed in basin. Nevertheless, the model's parameters showed a reliable behavior even in the presence of abrupt control actuation.

In relation to the identified parameters themselves, the inertia terms have roughly the same magnitude in a DOF, independent of the location or technique applied. The damping terms seems to have a combing effect. The linear and quadratic damping parameters can not be direct compared, since for some experiments the least-square method provided parameters with negative value, which would imply that these damping terms were applying energy to the system, instead of removing it. The reasons for such effect

would require more investigation. The buoyancy terms provided by both techniques are equivalent and each vehicle have a different value of buoyancy due a fine trim performed for the test location.

Considering the parameters identified for the four controllable DOF of FlatFish, one more experiment was considered in order to analyze how good was the model when performing a movement in four DOF at the same time with the vehicle. It was observed that neither the simplified model, in which the Coriolis effect is ignored, nor the complete model, in which the Coriolis effect is considered, are able to behave as the real vehicle when using the identified parameters of the simplified model. For the multi DOF actuation case, a parameters identification shall consider the complete model of the AUV.

Nevertheless, the complete model had a similar behavior with the Gazebo simulation when considering the method used to compensate for the added mass. It shows the effectiveness of the method proposed in 4.2.1. In case the parameters of the complete model would had been identified, the Gazebo simulation would behave as the real vehicle.

7.1 Future works

Future works include the addition of water current measurement to be considered during parameters identification. Additional tests to validate the method in the ocean, which presents water current, could be performed.

Methods of parameters identification that consider the complete dynamical model of an AUV should be studied and implemented. It is believed that with the complete model's parameters, the model, and the Gazebo simulation with the compensated effort, could behave as the real vehicle when submitted to the same control effort, even in the multi DOF actuation scenario.

A Savitzky-Golay filter

In 1964, Savitzky and Golay proposed a method to smooth and derivate a set of data based on least-square technique, [77]. The method, which is a Finite Impulse Response (FIR) lowpass filter [78], consist in determining an one-dimension convolutional kernel in order to fit a polynomial, or its derivative, in an evenly spaced dataset.

Previous works expanded the original formulation for computing the polynomial fit, and its derivatives, to all positions in the kernel, [79], and expanding the polynomial fitting to the multidimensional case, [80]. It's applied in fields like electroencephalographic [81], electrocardiography [82], image processing [83], [84] and analytical chemistry [85].

The one dimensional formulation used in this work is showed. Being a polynomial model $f(x)$ of order n given by

$$f(x) = \sum_{p=0}^n a_p x^p. \quad (\text{A.1})$$

Let the x coordinate of i^{th} data point be $x_i = i$, $-m \leq i \leq m$, being $(2m + 1)$ the window length, and the sampled value be g_i . The values of a_p are determined via least-square in order to fit $f(x_i)$ to g_i , i.e. by minimizing the total error

$$\epsilon = \sum_{i=-m}^m (f(x_i) - g_i)^2. \quad (\text{A.2})$$

The solution can be determine by taking the matrix form of

$$f(x_i) = g_i \quad (\text{A.3})$$

and use the pseudo-inverse

$$Pa = g \implies P^T Pa = P^T g \implies a = (P^T P)^{-1} P^T g, \quad (\text{A.4})$$

where

$$a = (a_0 \quad \dots \quad a_p \quad \dots \quad a_n)^T \quad (\text{A.5})$$

and P is the matrix which row multiplied by vector a gives the equivalent sample in vector g , defined by

$$P := \begin{pmatrix} 1 & x_{-m} & \dots & x_{-m}^p & \dots & x_{-m}^n \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_i & \dots & x_i^p & \dots & x_i^n \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_m & \dots & x_m^p & \dots & x_m^n \end{pmatrix}. \quad (\text{A.6})$$

P is usually a non-square matrix, with more rows than columns, since it is required to use a number of samples bigger than the order of the polynomial in order to avoid overfitting.

Writing the c^{th} derivative of Equation A.1 in terms of the coefficients a and the free variables as

$$\frac{\partial^c f}{\partial x^c} = \frac{\sum_{p=c}^n a_p \frac{p!}{(p-c)!} x^{p-c}}{\text{step}}, \quad (\text{A.7})$$

where the sampled data points are separated by step , [79], [80]. As the values on the right hand side are constants multiplying the values of a , defined in Equation A.4, A.7 can be written as

$$\frac{\partial^c f}{\partial x^c} = \frac{s(P^T P)^{-1} P^T g}{\text{step}}, \quad (\text{A.8})$$

where s is a vector with elements s_p defined by

$$s_p = \begin{cases} 0 & \text{if } 0 \leq p < c \\ \frac{p!}{(p-c)!} x^{p-c} & \text{if } c \leq p \leq n \end{cases} \quad (\text{A.9})$$

The dot product of $s(P^T P)^{-1} P^T$ with the vector of samples g and dividing by step , gives the c^{th} derivative at the required x value.

In order to exemplify the Savitzky-Golay filter, the first derivative of a signal with noise, specified in Table 15, is computed. The filter's parameters are: a polynomial of order 3, window size of 33 and first derivative related to the center point. The resultant convolutional kernel $s(P^T P)^{-1} P^T$ is given by Table 16.

The original signal as well as the computed derivative is shown in Figure 24. It's perceived the filter's capacity of computing the first derivative while removing the influence of noise.

Table 15 – Parameters of signal with noise

Signal	$0.3 * \sin(0.3 * t)$
Noise	$0.01 * \sin(10 * t) + 0.05 * \sin(3 * t)$
Step	0.25

Table 16 – Savitzky-Golay filter's kernel, polynomial order 3, 33 samples and first derivative at the center point

Index	-16	-15	-14	-13	-12
Coefficient	0.0107942	0.00507526	0.000332627	-0.00349878	-0.00648404
Index	-11	-10	-9	-8	-7
Coefficient	-0.00868824	-0.0101765	-0.0110138	-0.0112654	-0.0109963
Index	-6	-5	-4	-3	-2
Coefficient	-0.0102715	-0.00915624	-0.00771552	-0.00601445	-0.00411811
Index	-1	0	1	2	3
Coefficient	-0.0020916	0	0.0020916	0.00411811	0.00601445
Index	4	5	6	7	8
Coefficient	0.00771552	0.00915624	0.0102715	0.0109963	0.0112654
Index	9	10	11	12	13
Coefficient	0.0110138	0.0101765	0.00868824	0.00648404	0.00349878
Index	14	15	16		
Coefficient	-0.000332627	-0.00507526	-0.0107942		

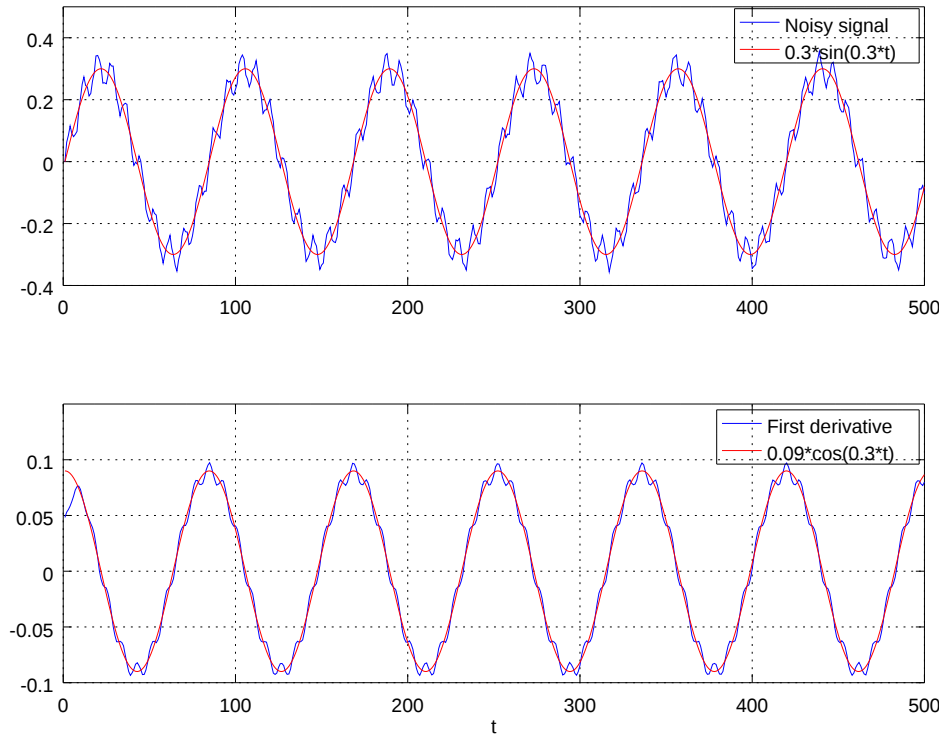


Figure 24 – Signal with noise and its first derivative computed by Savitzky-Golay filter

B Parameters Identification Dataset

This appendix shows the dataset collect during identification experiments, as well as the model's parameters and velocity error provided by the adaptive identifier.

B.1 Basin

B.1.1 Surge

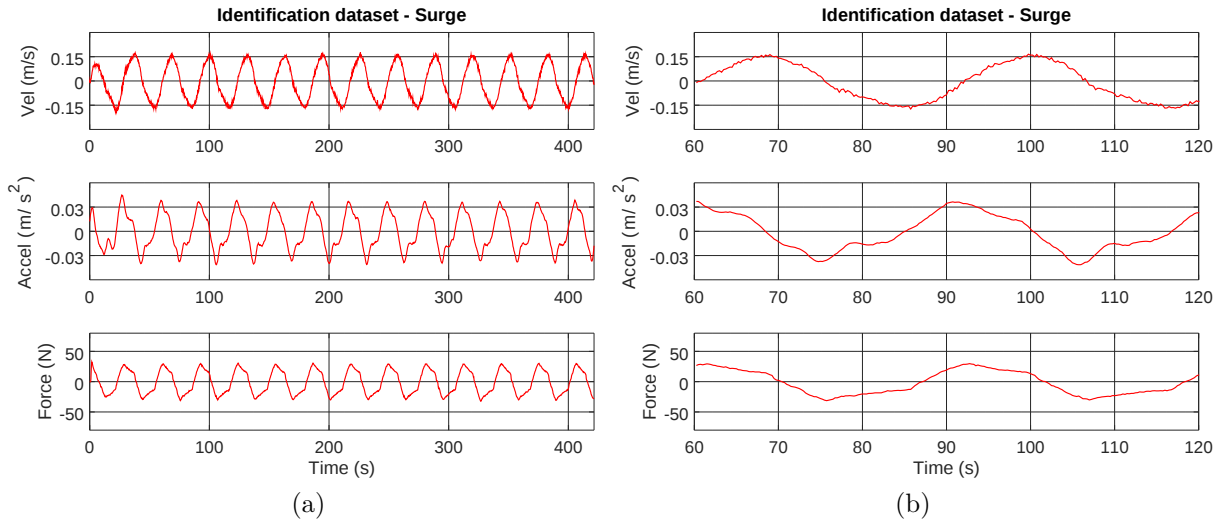


Figure 25 – Identification dataset for surge, done in basin: (a) Full trial, (b) 60s zoom in

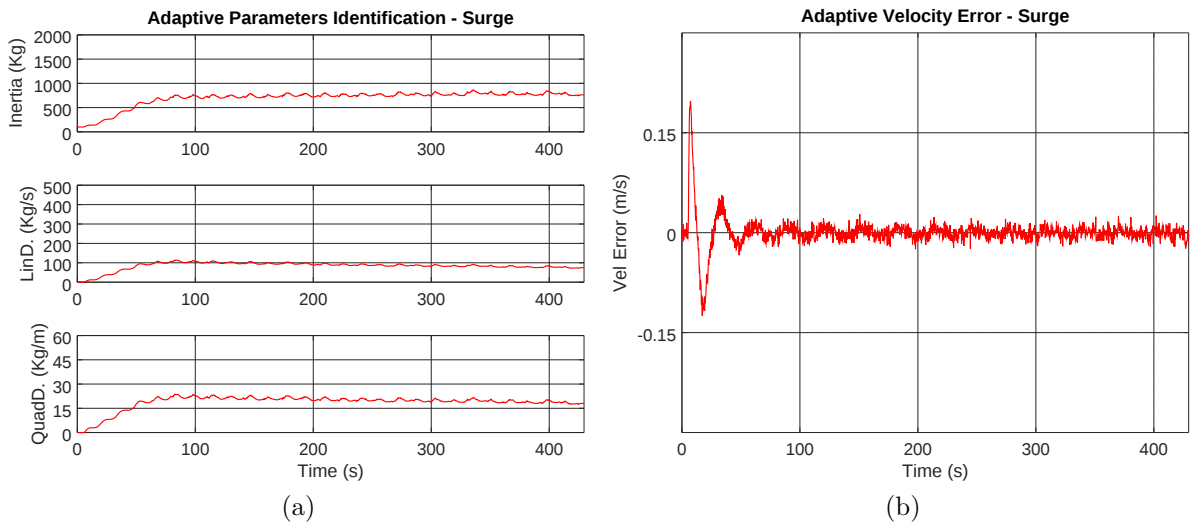


Figure 26 – Output of adaptive identifier for surge. $a_m = 1$, $\lambda_1 = 2e - 4$, $\lambda_2 = 5e - 1$, $\lambda_3 = 5e - 1$, $\lambda_4 = 0$: (a) Parameters, (b) Velocity error

B.1.2 Sway

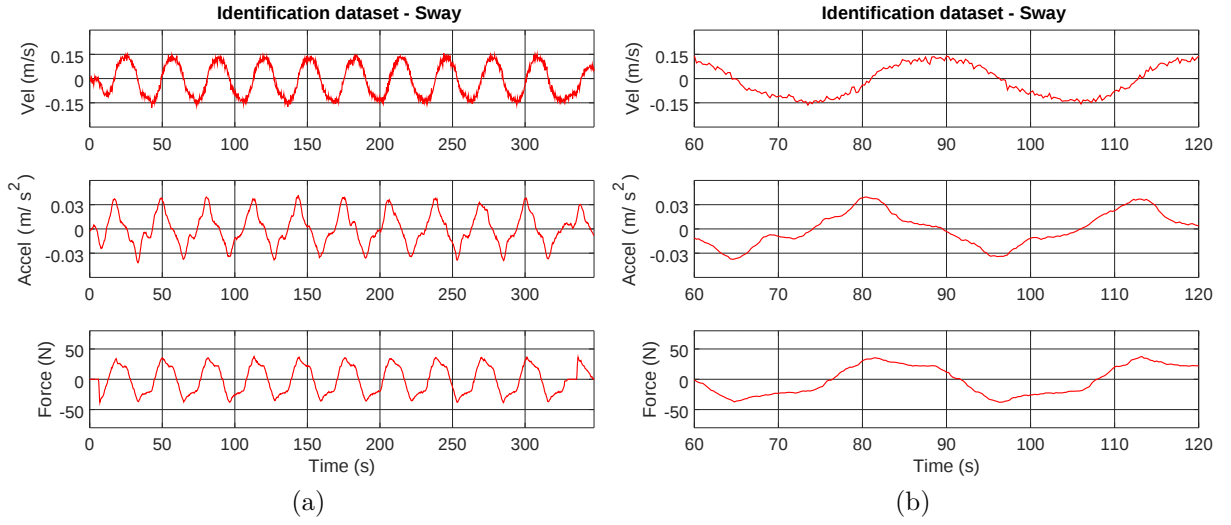
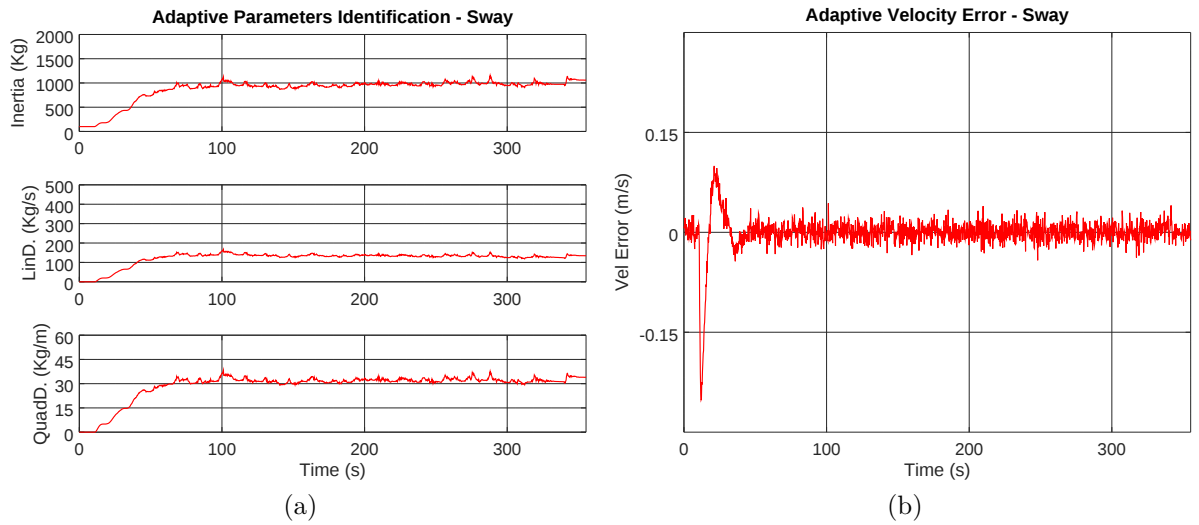


Figure 27 – Identification dataset for sway, done in basin: (a) Full trial, (b) 60s zoom in

Figure 28 – Output of adaptive identifier for sway. $a_m = 1$, $\lambda_1 = 2e - 4$, $\lambda_2 = 5e - 1$, $\lambda_3 = 5e - 1$, $\lambda_4 = 0$: (a) Parameters, (b) Velocity error

B.1.3 Heave

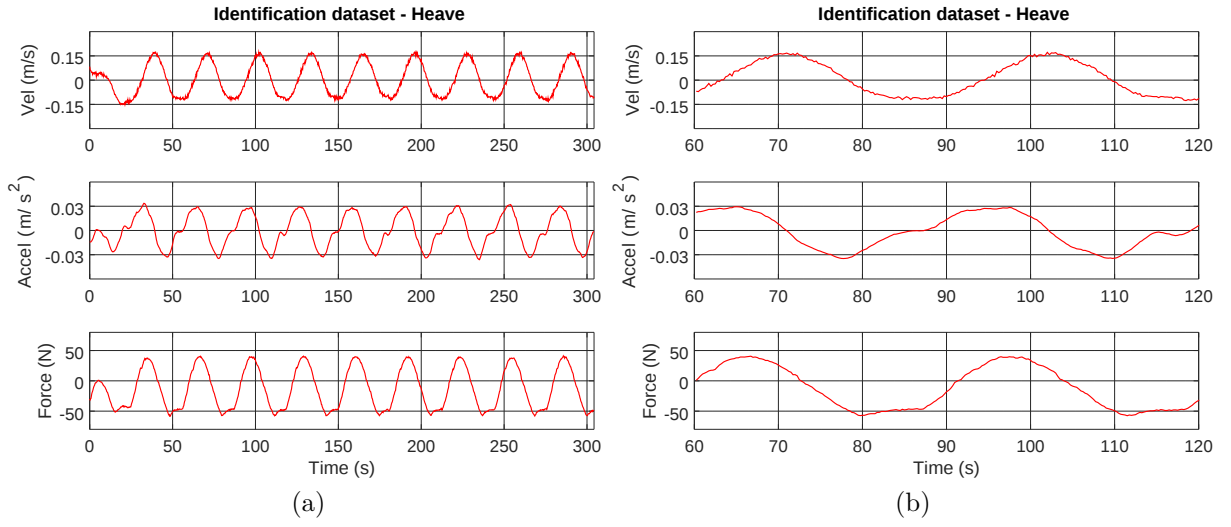
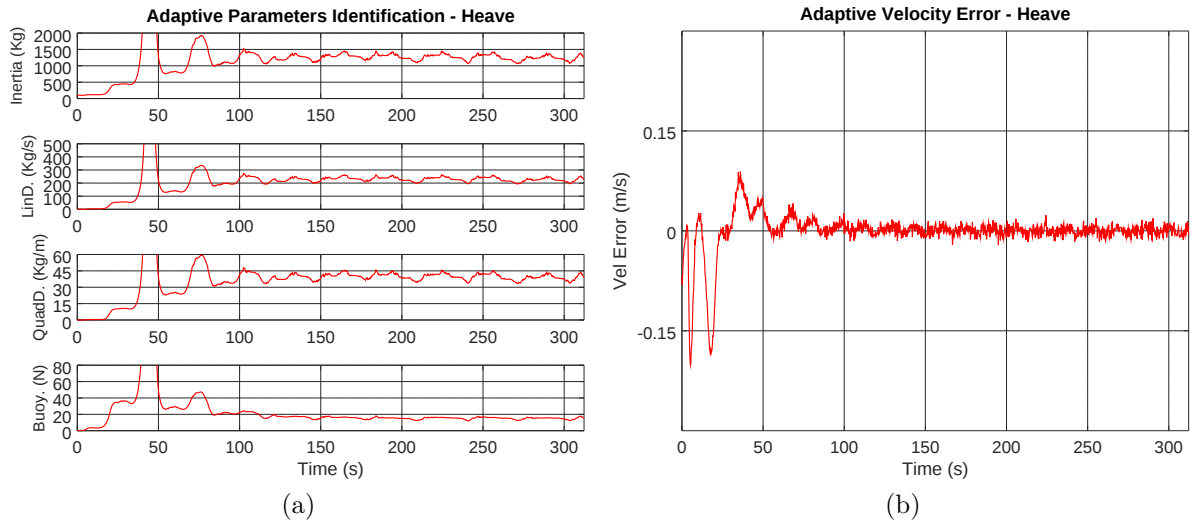


Figure 29 – Identification dataset for heave, done in basin: (a) Full trial, (b) 60s zoom in

Figure 30 – Output of adaptive identifier for heave. $a_m = 1$, $\lambda_1 = 1.5e-4$, $\lambda_2 = 5e-1$, $\lambda_3 = 5e-1$, $\lambda_4 = 5e-2$: (a) Parameters, (b) Velocity error

B.1.4 Yaw

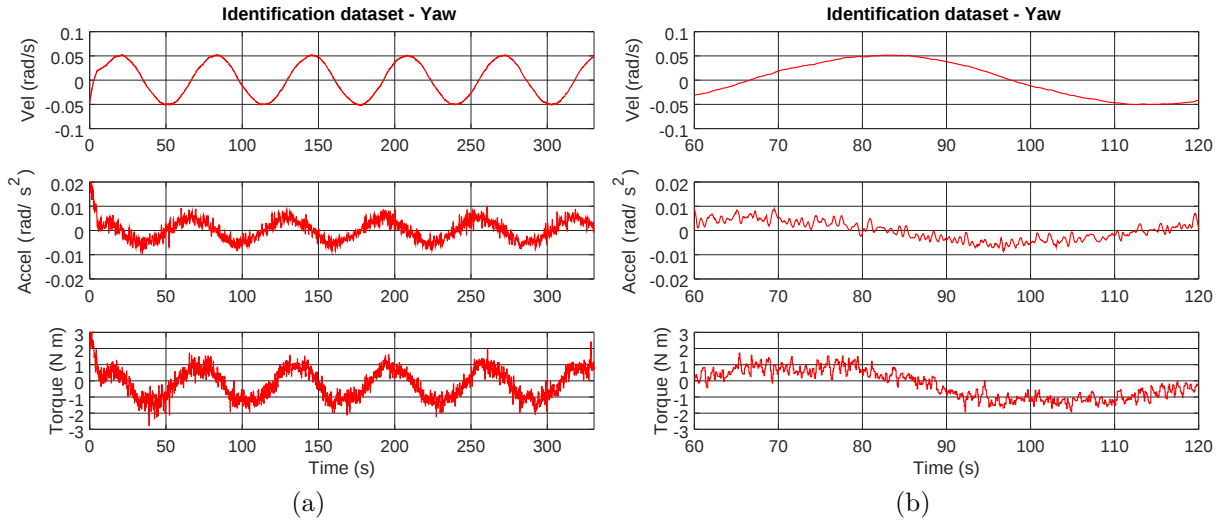
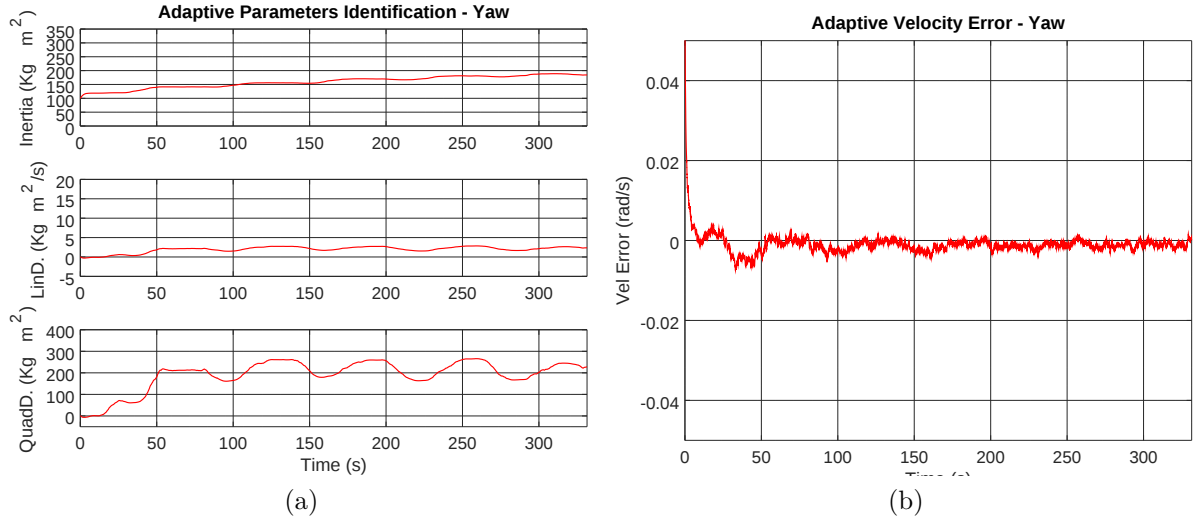


Figure 31 – Identification dataset for yaw, done in basin: (a) Full trial, (b) 60s zoom in

Figure 32 – Output of adaptive identifier for yaw. $a_m = 1$, $\lambda_1 = 1e-2$, $\lambda_2 = 1e4$, $\lambda_3 = 5e0$, $\lambda_4 = 0$: (a) Parameters, (b) Velocity error

B.2 Bay

B.2.1 Surge

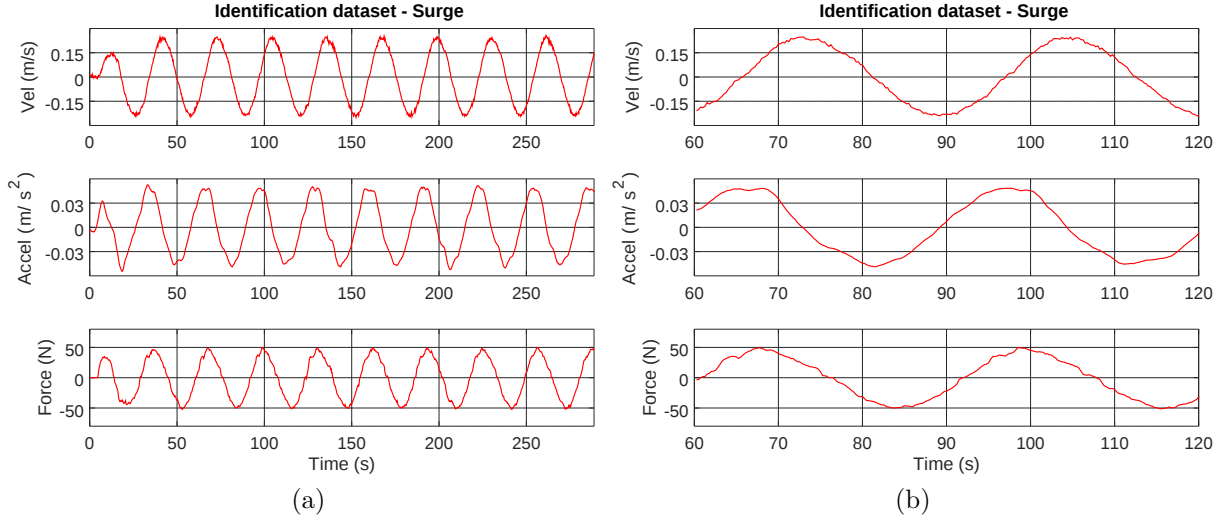


Figure 33 – Identification dataset for surge, done in bay: (a) Full trial, (b) 60s zoom in

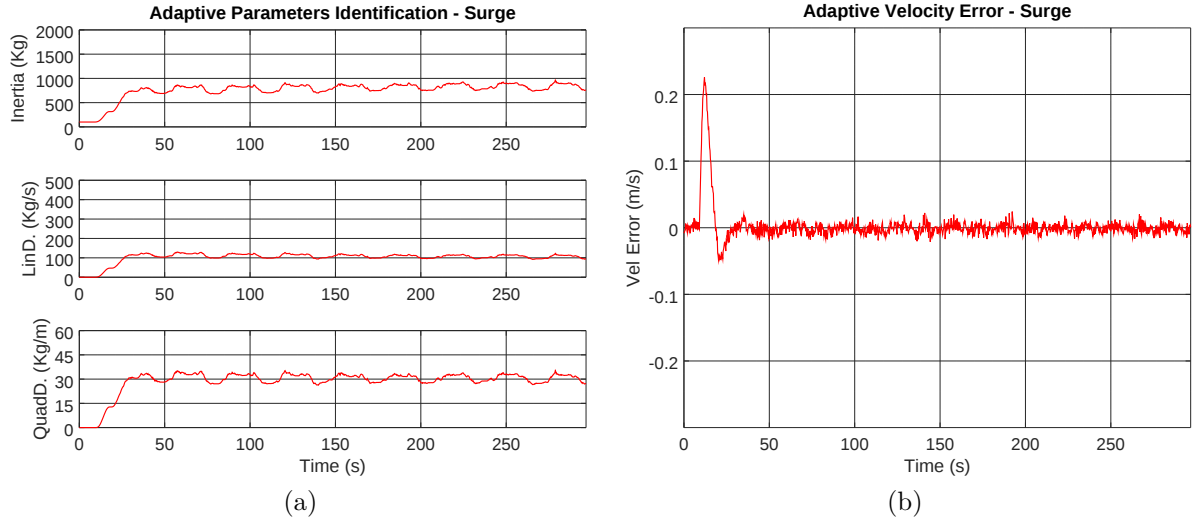


Figure 34 – Output of adaptive identifier for surge. $a_m = 1$, $\lambda_1 = 2e - 4$, $\lambda_2 = 5e - 1$, $\lambda_3 = 5e - 1$, $\lambda_4 = 0$: (a) Parameters, (b) Velocity error

B.2.2 Sway

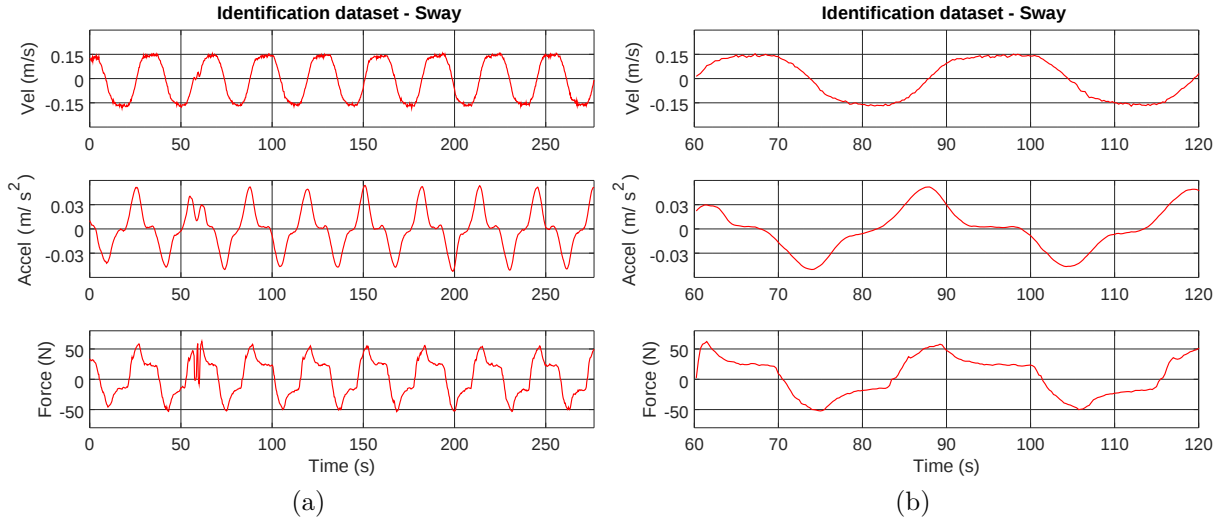
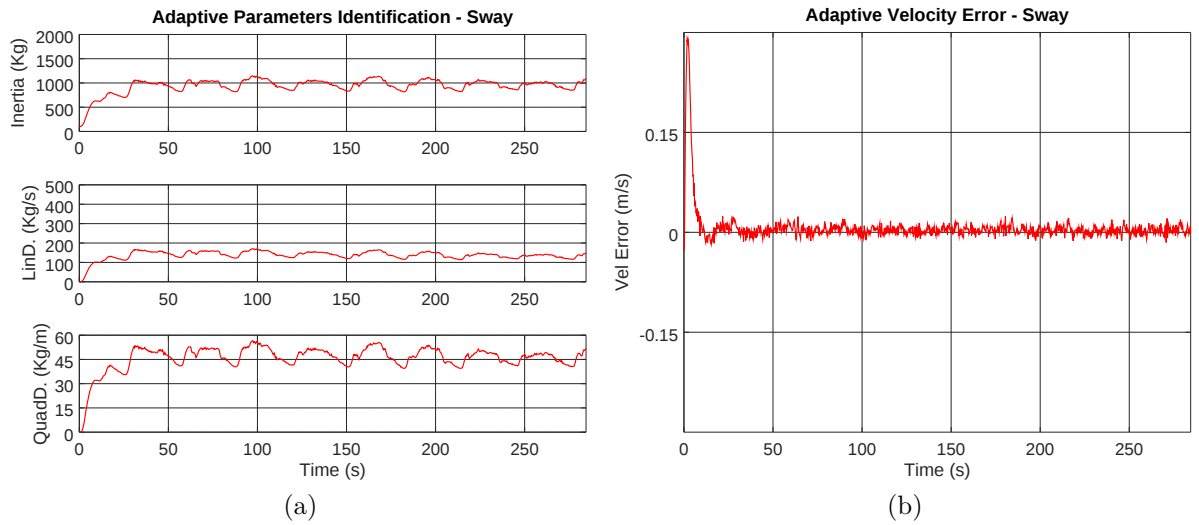


Figure 35 – Identification dataset for sway, done in bay: (a) Full trial, (b) 60s zoom in

Figure 36 – Output of adaptive identifier for sway. $a_m = 1$, $\lambda_1 = 2e - 4$, $\lambda_2 = 5e - 1$, $\lambda_3 = 5e - 1$, $\lambda_4 = 0$: (a) Parameters, (b) Velocity error

B.2.3 Heave

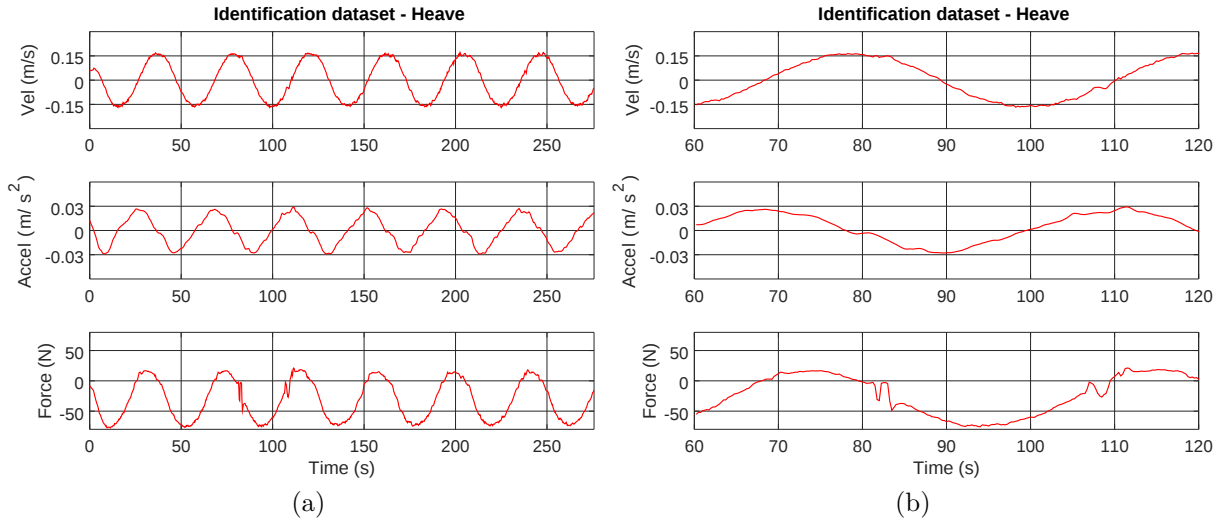
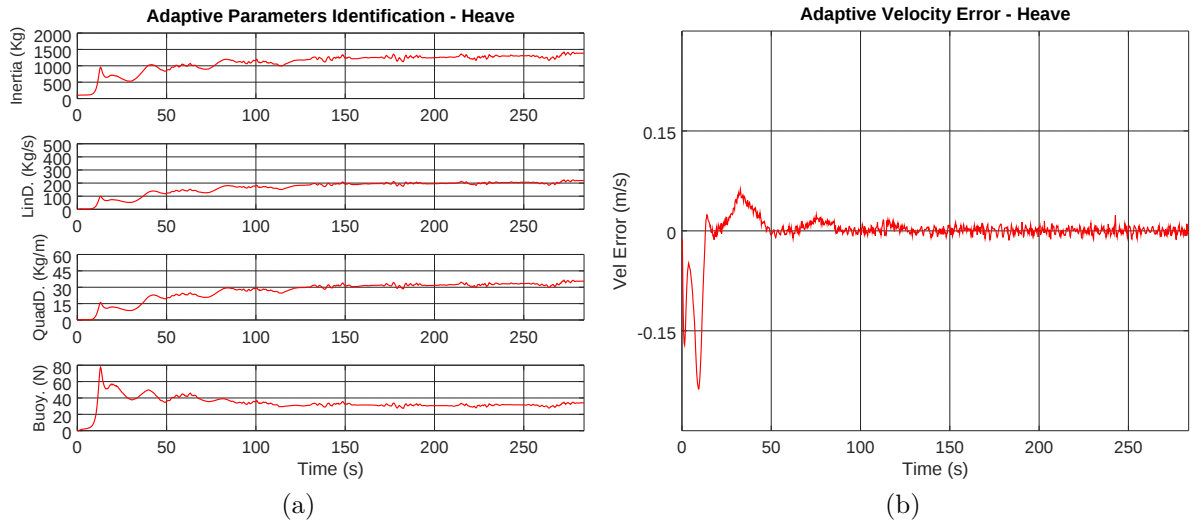


Figure 37 – Identification dataset for heave, done in bay: (a) Full trial, (b) 60s zoom in

Figure 38 – Output of adaptive identifier for heave. $a_m = 1$, $\lambda_1 = 1.5e-4$, $\lambda_2 = 5e-1$, $\lambda_3 = 5e-1$, $\lambda_4 = 5e-2$: (a) Parameters, (b) Velocity error

B.2.4 Yaw

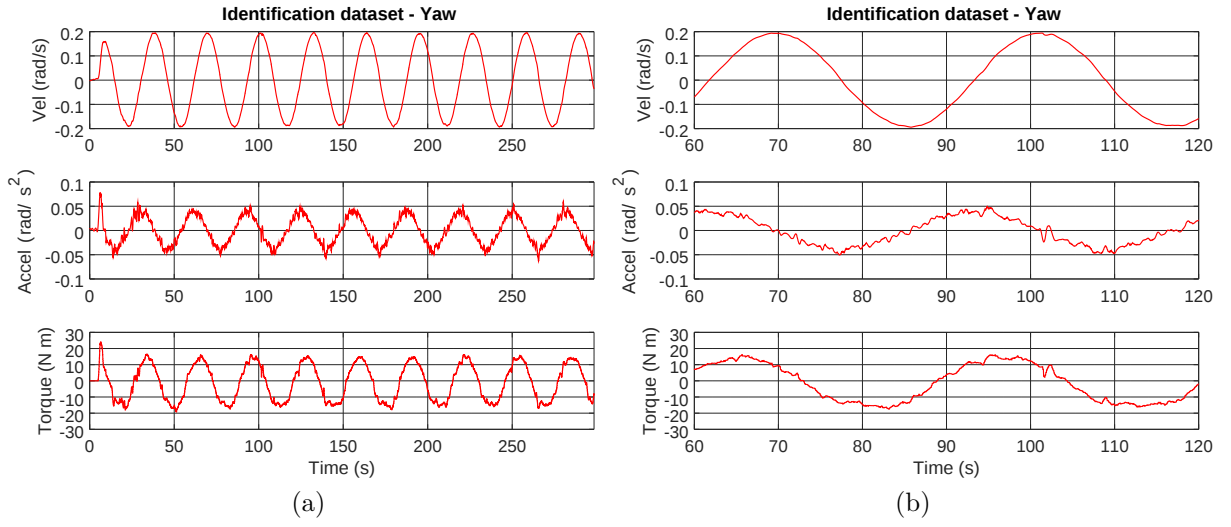
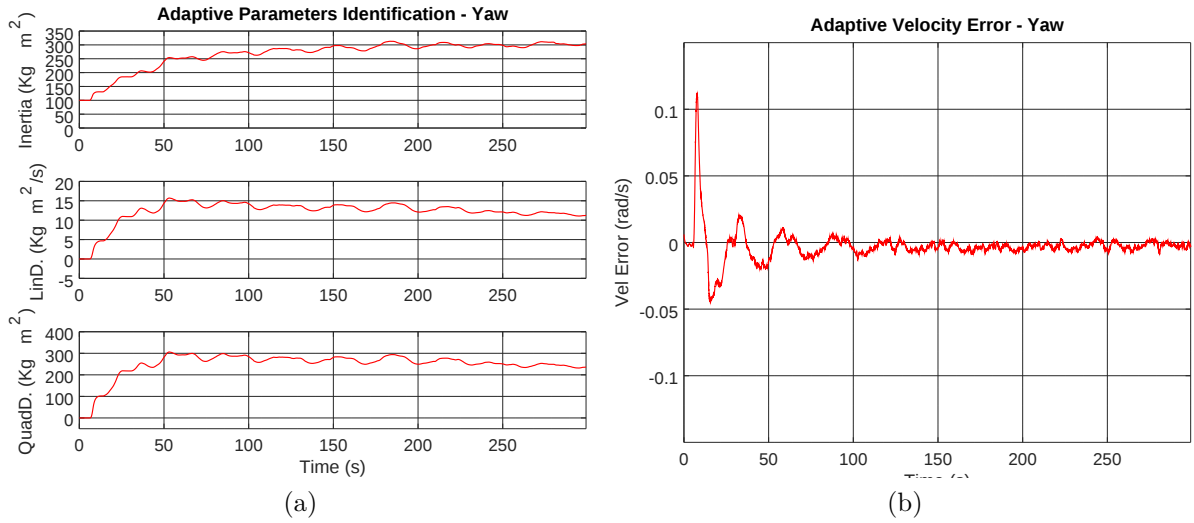


Figure 39 – Identification dataset for yaw, done in bay: (a) Full trial, (b) 60s zoom in

Figure 40 – Output of adaptive identifier for yaw. $a_m = 1$, $\lambda_1 = 5e - 4$, $\lambda_2 = 5e1$, $\lambda_3 = 5e - 1$, $\lambda_4 = 0$: (a) Parameters, (b) Velocity error

C Influence of Gain in Adaptive Identifier

The adaptive identifier has several gain parameters to be empirically tuned. This appendix explores the influence of two gains, a and λ_1 , in the parameters identification and its performance. The base case for this analyses was the dataset of surge DOF performed in the basin. Surge identification dataset, in Table 5, was used for parameters identification, while the surge cross-validation dataset, in Table 6 was used for validation of the identified parameters.

Five values of gain a were used, being $[0.1, 0.5, 1, 1.5, 2]$, while for the gain λ_1 , one hundred values were used, from 0.00001 to 0.001. Five hundreds sets of model's parameters were identified and their mean absolute error with the cross-validation dataset were performed.

Figures 41, 42, 43, 44 and 45 present the time evolution of the identified parameters with the velocity error as a function of λ_1 for a equal to 0.1, 0.5, 1, 1.5 and 2 respectively. The last parameters in time provided by the adaptive identifiers, i.e. the final parameters as a function of λ_1 , are presented in Figures 46, 47, 48, 49 and 50 for a equal to 0.1, 0.5, 1, 1.5 and 2 respectively, altogether with the MAE for the correspondent model's parameters and the MAE for the parameters provided by the least square method, present in Table 8, shown as a reference value.

For gain $a = 0.1$, in Figure 41, the adaptive identifier performs poorly, with abrupt changes in parameters values. From Figures 42, 43, 44 and 45 it's perceived that for high values of λ_1 , the inertia parameter provided by the adaptive identifier becomes highly oscillatory and for λ_1 smaller than 0.0001 the inertia term slowly changes its value while the damping terms increase value quickly. It is also perceived that for $\lambda_1 > 0.0001$, the velocity error of the adaptive identifier tends to zero faster. Increasing the value of a causes the velocity error to converge to zero faster and reduce the oscillation's amplitude of the parameters.

Although the Figures 46, 47, 48, 49, 50 may suggest that the parameters converge by increasing the values of a and λ_1 , they represent the last parameters provided by the adaptive identifiers, with by chance have similar values. In case the last samples were placed in a parameter's peak, they would be considerable different, since no smooth filter was applied in the adaptive identifier's output.

Nevertheless, the model's MAE when using the identified parameters via adaptive identifier is mostly similar with the model configured with parameters identified via the least-square method. A minimal value of MAE can be found for a gain λ_1 around 0.0002. The time evolution of parameters present in Figure 26, with $a = 1$ and $\lambda_1 = 0.0002$, is

present in Figure 43 for $\lambda_1 = 0.0002$.

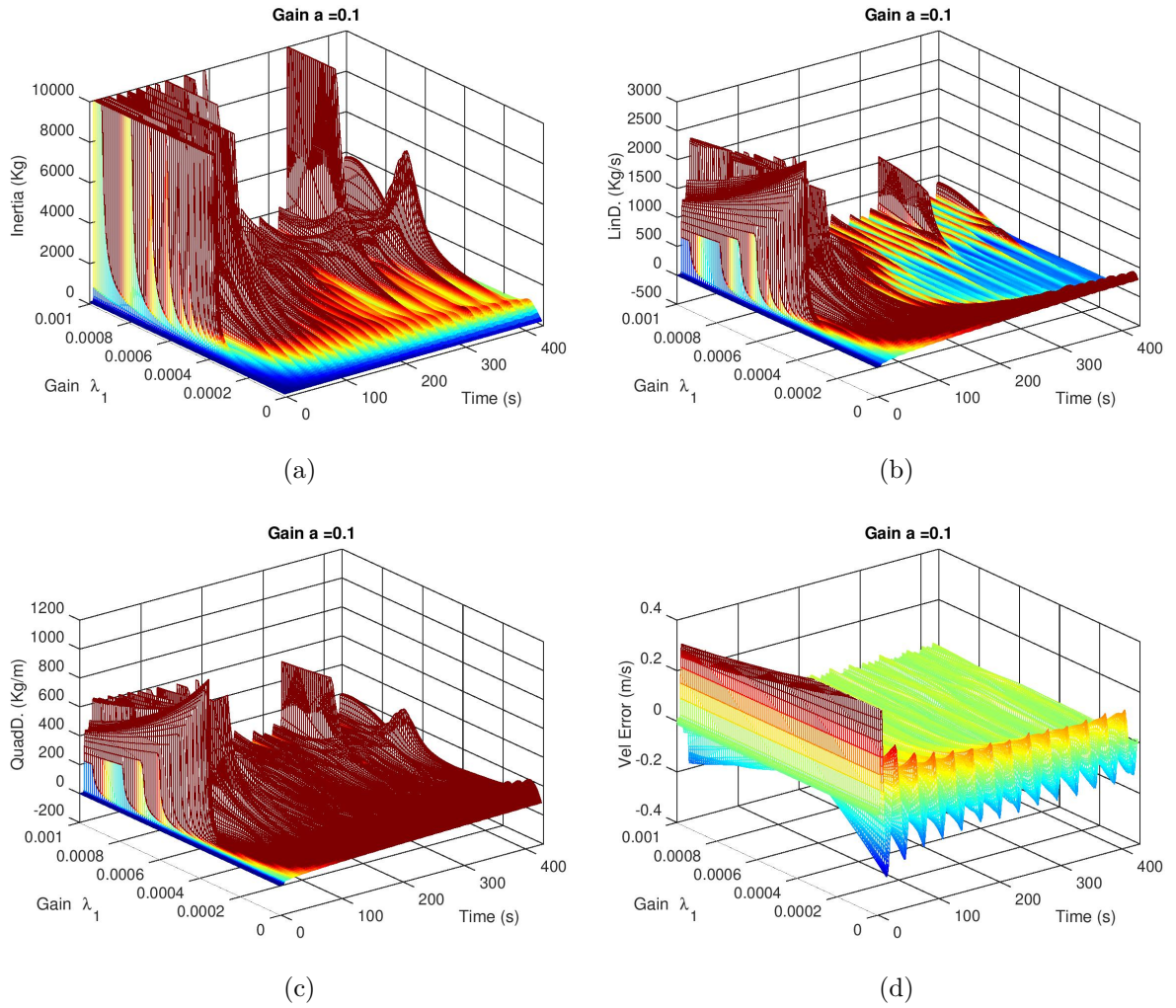


Figure 41 – Output over timer of adaptive identifier in surge DOF as function of λ_1 , $a = 0.1$
: (a) Inertia (b) Linear Damping, (c) Quadratic Damping, (d) Velocity Error

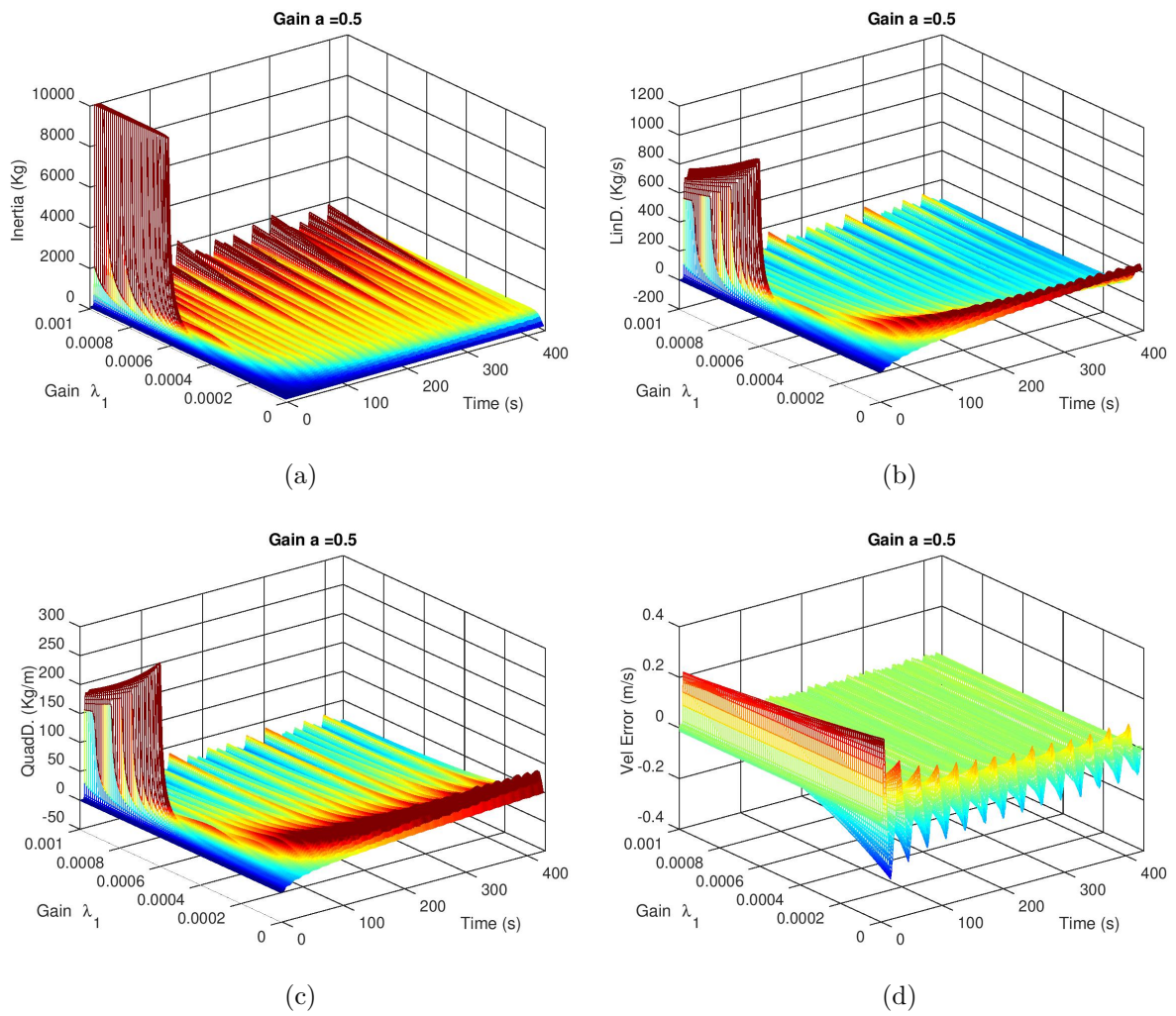


Figure 42 – Output over timer of adaptive identifier in surge DOF as function of λ_1 , $a = 0.5$: (a) Inertia (b) Linear Damping, (c) Quadratic Damping, (d) Velocity Error

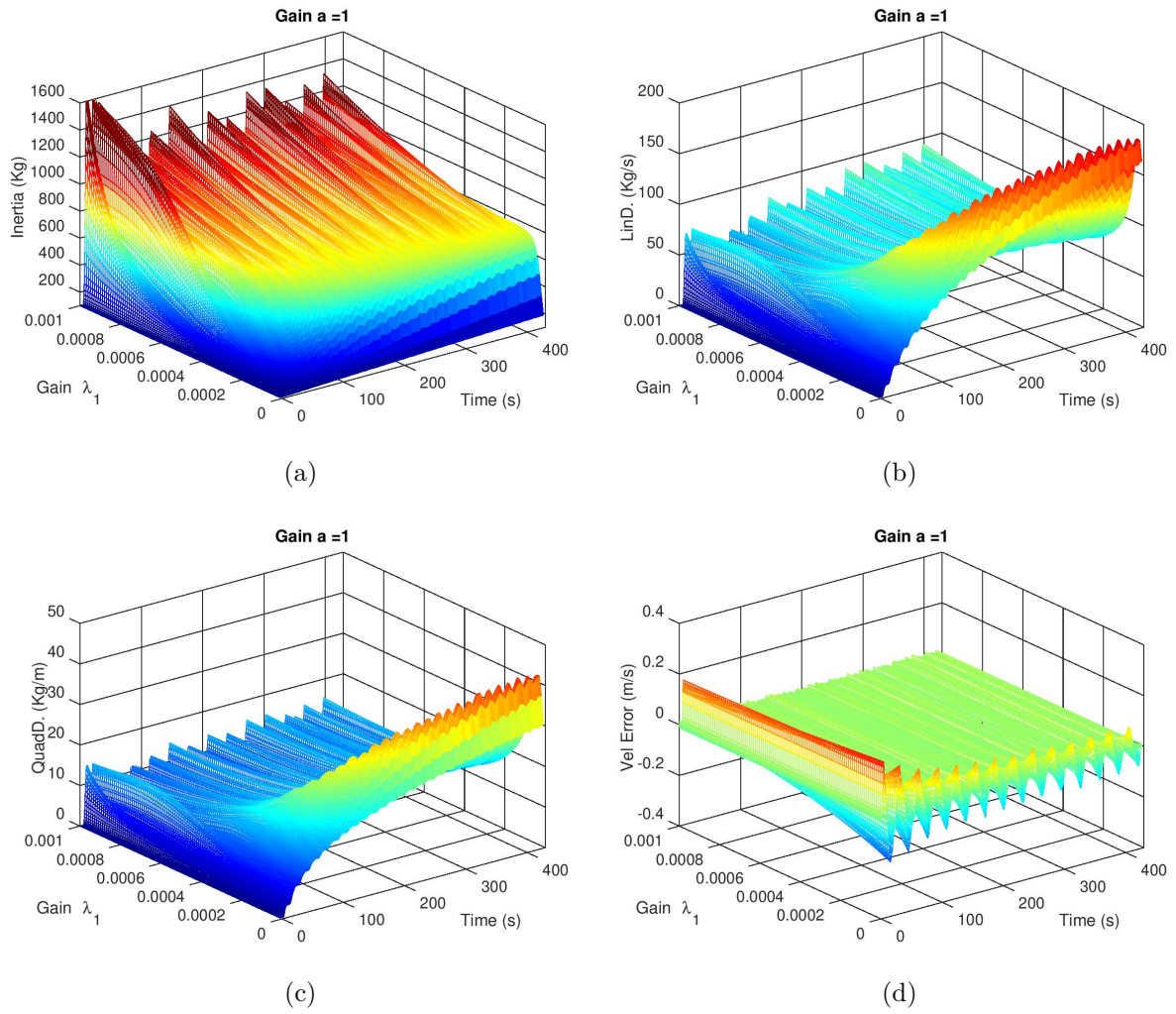


Figure 43 – Output over timer of adaptive identifier in surge DOF as function of λ_1 , $a = 1$: (a) Inertia (b) Linear Damping, (c) Quadratic Damping, (d) Velocity Error

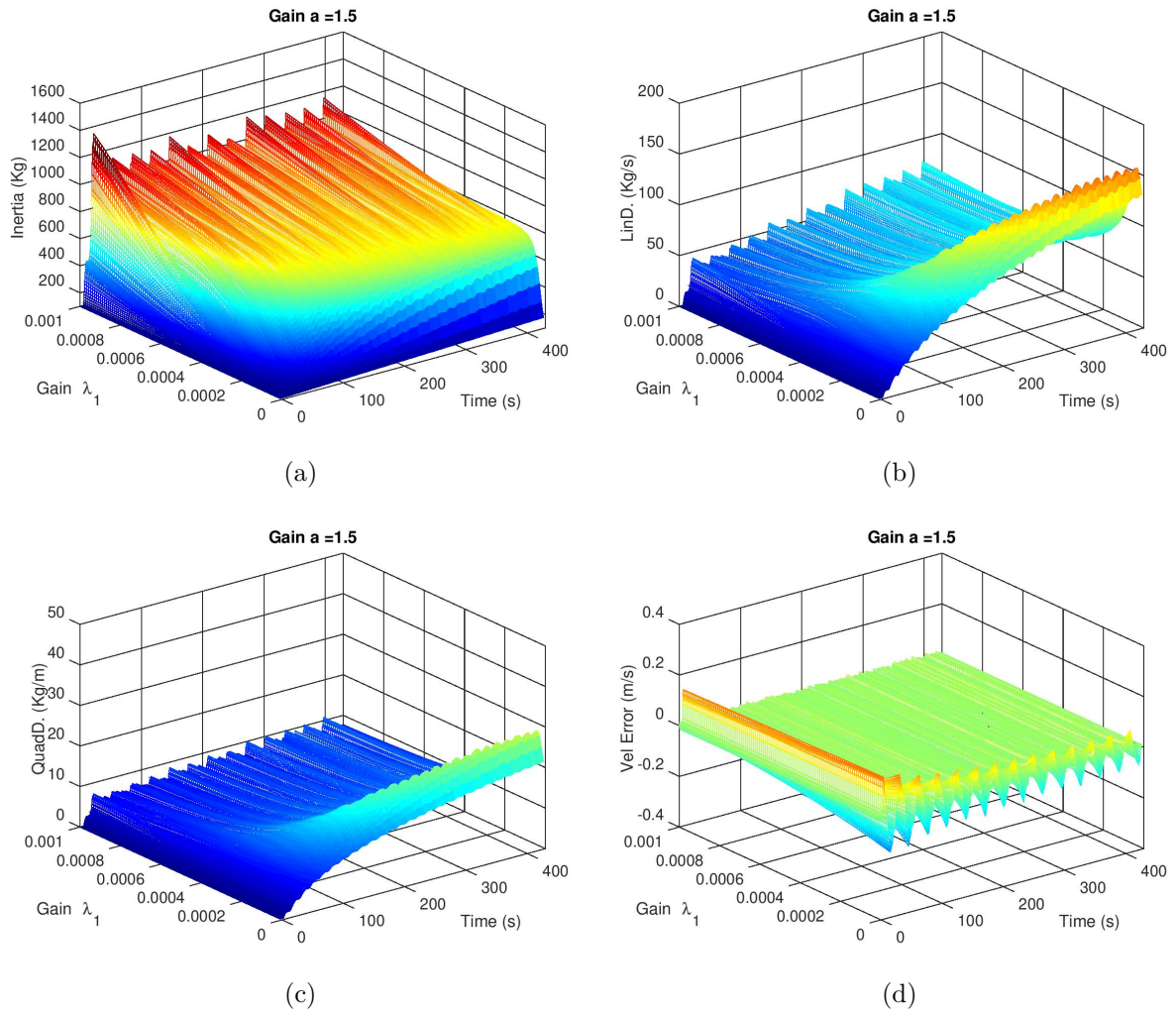


Figure 44 – Output over timer of adaptive identifier in surge DOF as function of λ_1 , $a = 1.5$: (a) Inertia (b) Linear Damping, (c) Quadratic Damping, (d) Velocity Error

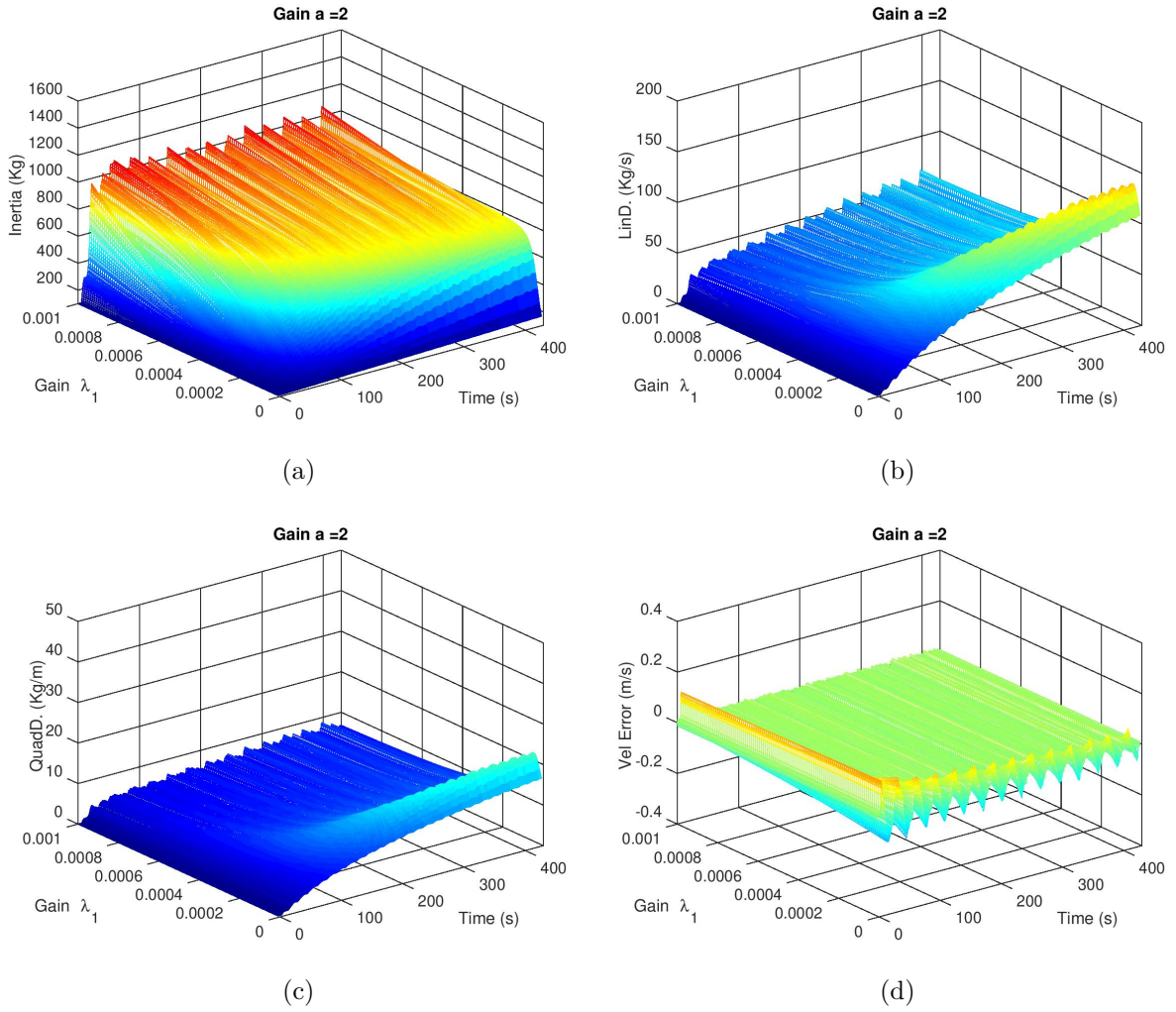


Figure 45 – Output over timer of adaptive identifier in surge DOF as function of λ_1 , $a = 2$: (a) Inertia (b) Linear Damping, (c) Quadratic Damping, (d) Velocity Error

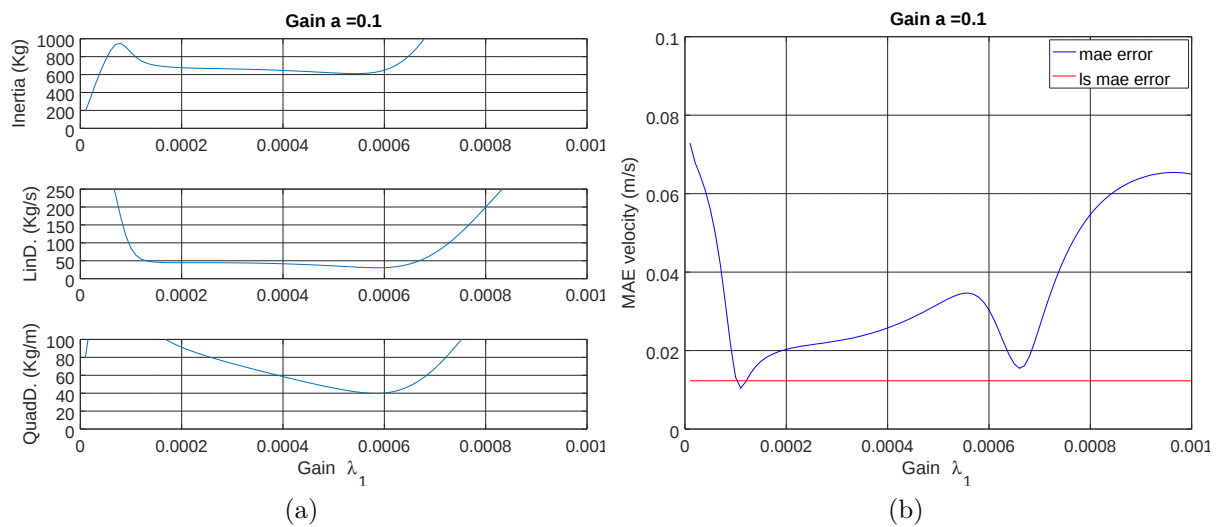


Figure 46 – Variation of gain λ_1 in adaptive identifier, with gain $a = 0.1$: (a) Identified parameters, using identification dataset (b) Mean absolute error of identified parameters, using cross-validation dataset

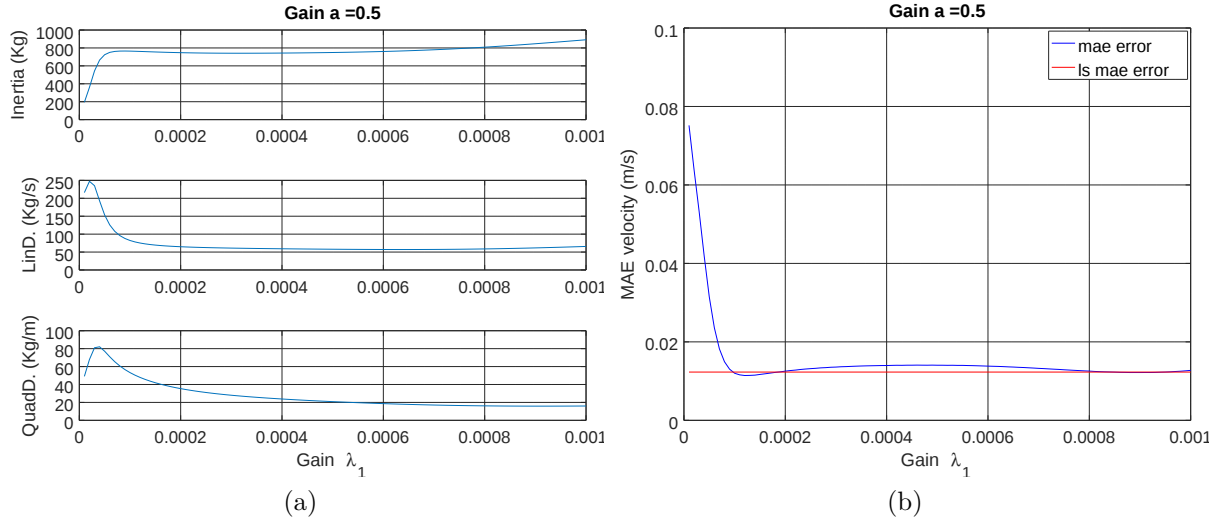


Figure 47 – Variation of gain λ_1 in adaptive identifier, with gain $a = 0.5$: (a) Identified parameters, using identification dataset (b) Mean absolute error of identified parameters, using cross-validation dataset

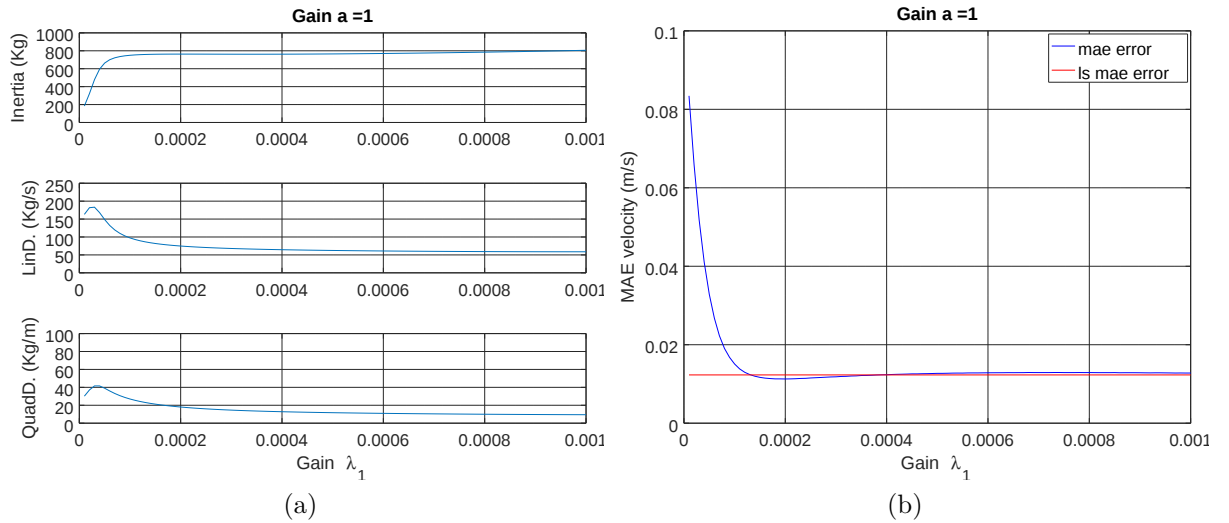


Figure 48 – Variation of gain λ_1 in adaptive identifier, with gain $a = 1$: (a) Identified parameters, using identification dataset (b) Mean absolute error of identified parameters, using cross-validation dataset

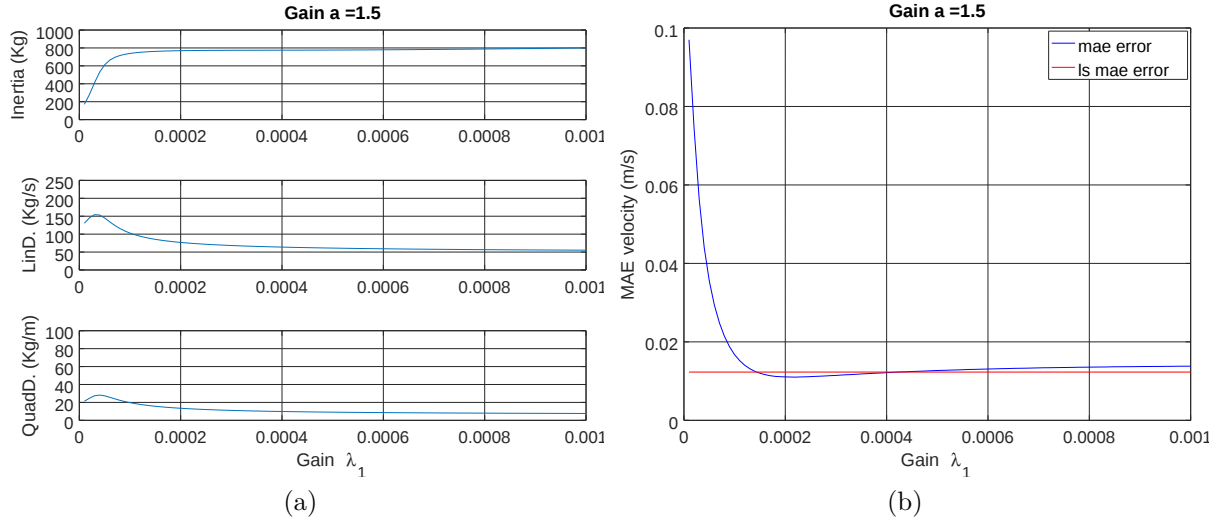


Figure 49 – Variation of gain λ_1 in adaptive identifier, with gain $a = 1.5$: (a) Identified parameters, using identification dataset (b) Mean absolute error of identified parameters, using cross-validation dataset

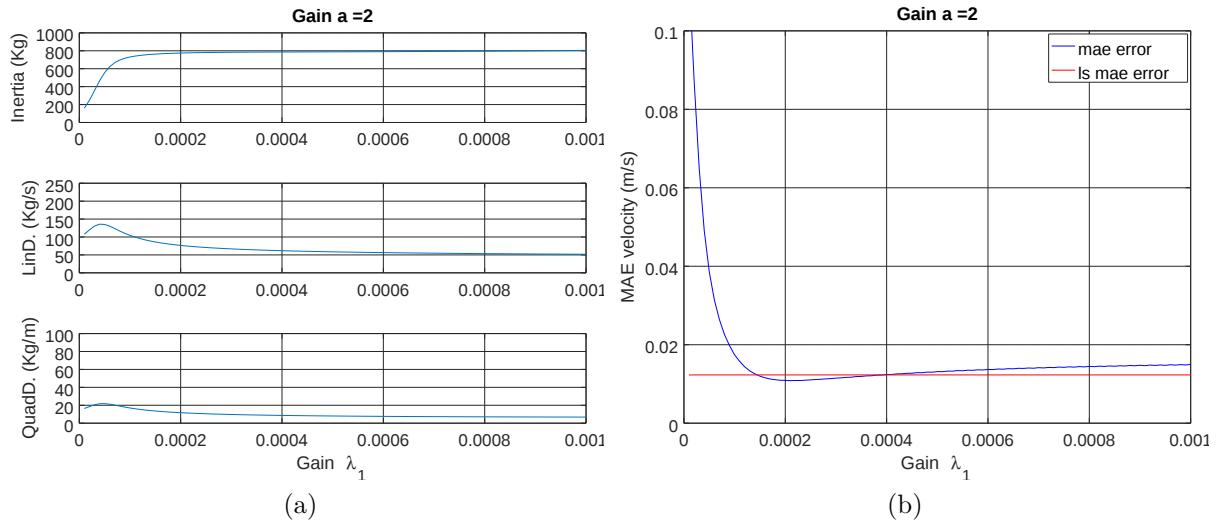


Figure 50 – Variation of gain λ_1 in adaptive identifier, with gain $a = 2$: (a) Identified parameters, using identification dataset (b) Mean absolute error of identified parameters, using cross-validation dataset

Bibliography

- 1 ALBIEZ, J. et al. FlatFish – a compact subsea-resident inspection auv. In: *Proceedings of the MTS/IEEE OCEANS 2015*. Washington DC, USA: [s.n.], 2015. p. 1–8. Cited 6 times on pages 8, 11, 16, 20, 21, and 25.
- 2 SABACK, R. et al. Fault-tolerant control allocation technique based on explicit optimization applied to an autonomous underwater vehicle. In: IEEE. *OCEANS 2016 MTS/IEEE Monterey*. [S.l.], 2016. p. 1–8. Cited 4 times on pages 8, 22, 25, and 27.
- 3 Enitech. *Rim-Driven Thruster*. 2014. Available from: <http://enitech.de/files/produkte/Datasheet_Rim-Driven_Thruster.pdf>. Accessed: 2018-04-13. Cited 2 times on pages 8 and 23.
- 4 SABACK, R. Robust nonlinear model predictive control with fault-tolerant control allocation applied to an autonomous underwater vehicle. Escola Politécnica, 2017. Cited 3 times on pages 8, 17, and 26.
- 5 TP, S. B.; KUMAR, S. Underwater communications. In: IEEE. *Underwater Technology (UT), 2015 IEEE*. [S.l.], 2015. p. 1–3. Cited on page 16.
- 6 QUATTRINI, A. M.; DEMOPOULOS, A. W. Ectoparasitism on deep-sea fishes in the western north atlantic: In situ observations from rovs surveys. *International Journal for Parasitology: Parasites and Wildlife*, Elsevier, v. 5, n. 3, p. 217–228, 2016. Cited on page 16.
- 7 BINGHAM, B. et al. Robotic tools for deep water archaeology: Surveying an ancient shipwreck with an autonomous underwater vehicle. *Journal of Field Robotics*, Wiley Online Library, v. 27, n. 6, p. 702–717, 2010. Cited on page 16.
- 8 HAGEN, P. E.; STORKERSEN, N. J.; VESTGARD, K. Hugin-use of uuv technology in marine applications. In: IEEE. *OCEANS'99 MTS/IEEE. Riding the Crest into the 21st Century*. [S.l.], 1999. v. 2, p. 967–972. Cited on page 16.
- 9 RATMEYER, V.; RIGAUD, V. Europe's growing fleet of scientific deepwater rovs: Emerging demands for interchange, workflow enhancement and training. In: IEEE. *OCEANS 2009-EUROPE*. [S.l.], 2009. p. 1–6. Cited on page 16.
- 10 ZAGATTI, R. et al. Flatfish resident auv: Leading the autonomy era for subsea oil and gas operations. In: OFFSHORE TECHNOLOGY CONFERENCE. *Offshore Technology Conference*. [S.l.], 2018. Cited on page 16.
- 11 GILMOUR, B.; NICCUM, G.; O'DONNELL, T. Field resident auv systems—chevron's long-term goal for auv development. In: IEEE. *Autonomous Underwater Vehicles (AUV), 2012 IEEE/OES*. [S.l.], 2012. p. 1–5. Cited on page 16.
- 12 SIESJÖ, J.; ROPER, C.; FURUHOLMEN, M. Sabertooth a seafloor resident hybrid auv/rov system for long term deployment in deep water and hostile environments. *UUST 2013*, 2013. Cited on page 16.

- 13 KOENIG, N.; HOWARD, A. Design and use paradigms for gazebo, an open-source multi-robot simulator. In: IEEE. *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*. [S.l.], 2004. v. 3, p. 2149–2154. Cited 2 times on pages 17 and 33.
- 14 KERMORGANT, O. A dynamic simulator for underwater vehicle-manipulators. In: SPRINGER. *International Conference on Simulation, Modeling, and Programming for Autonomous Robots*. [S.l.], 2014. p. 25–36. Cited 3 times on pages 17, 34, and 35.
- 15 MANHÃES, M. M. M. et al. Uuv simulator: A gazebo-based package for underwater intervention and multi-robot simulation. In: IEEE. *OCEANS 2016 MTS/IEEE Monterey*. [S.l.], 2016. p. 1–8. Cited 3 times on pages 17, 34, and 35.
- 16 PRATS, M. et al. An open source tool for simulation and supervision of underwater intervention missions. In: IEEE. *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. [S.l.], 2012. p. 2577–2582. Cited 2 times on pages 17 and 34.
- 17 CACCIA, M.; INDIVERI, G.; VERUGGIO, G. Modeling and identification of open-frame variable configuration unmanned underwater vehicles. *Oceanic Engineering, IEEE Journal of*, v. 25, n. 2, p. 227–240, April 2000. ISSN 0364-9059. Cited 3 times on pages 17, 30, and 43.
- 18 SMALLWOOD, D.; WHITCOMB, L. Adaptive identification of dynamically positioned underwater robotic vehicles. *Control Systems Technology, IEEE Transactions on*, v. 11, n. 4, p. 505–515, July 2003. ISSN 1063-6536. Cited 5 times on pages 17, 31, 43, 44, and 45.
- 19 RIDAO, P. et al. On the identification of non-linear models of unmanned underwater vehicles. *Control Engineering Practice*, v. 12, n. 12, p. 1483 – 1499, 2004. ISSN 0967-0661. Guidance and control of underwater vehicles. Cited 2 times on pages 17 and 43.
- 20 MCFARLAND, C.; WHITCOMB, L. Comparative experimental evaluation of a new adaptive identifier for underwater vehicles. In: *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. [S.l.: s.n.], 2013. p. 4614–4620. ISSN 1050-4729. Cited 4 times on pages 17, 30, 36, and 43.
- 21 SMALLWOOD, D. A.; WHITCOMB, L. L. Model-based dynamic positioning of underwater robotic vehicles: theory and experiment. *IEEE Journal of Oceanic Engineering*, IEEE, v. 29, n. 1, p. 169–186, 2004. Cited on page 17.
- 22 CHE, X. et al. Re-evaluation of rf electromagnetic communication in underwater sensor networks. *IEEE Communications Magazine*, IEEE, v. 48, n. 12, p. 143–151, 2010. Cited on page 17.
- 23 WHITCOMB, L. et al. Towards precision robotic maneuvering, survey, and manipulation in unstructured undersea environments. In: *Robotics Research*. [S.l.]: Springer, 1998. p. 45–54. Cited on page 18.
- 24 HEGRENAES, O.; BERGLUND, E.; HALLINGSTAD, O. Model-aided inertial navigation for underwater vehicles. In: *ICRA*. [S.l.: s.n.], 2008. p. 1069–1076. Cited on page 18.

- 25 KINSEY, J. C.; YANG, Q.; HOWLAND, J. C. Nonlinear dynamic model-based state estimators for underwater navigation of remotely operated vehicles. *IEEE Transactions on Control Systems Technology*, IEEE, v. 22, n. 5, p. 1845–1854, 2014. Cited on page 18.
- 26 FRANK, P. M. Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy: A survey and some new results. *automatica*, Elsevier, v. 26, n. 3, p. 459–474, 1990. Cited on page 18.
- 27 ALESSANDRI, A.; CACCIA, M.; VERUGGIO, G. Fault detection of actuator faults in unmanned underwater vehicles. *Control Engineering Practice*, Pergamon, v. 7, n. 3, p. 357–368, 1999. Cited on page 18.
- 28 ISERMANN, R. Process fault detection based on modeling and estimation methods—a survey. *automatica*, Elsevier, v. 20, n. 4, p. 387–404, 1984. Cited on page 18.
- 29 YOERGER, D.; COOKE, J.; SLOTINE, J.-J. The influence of thruster dynamics on underwater vehicle behavior and their incorporation into control system design. *Oceanic Engineering, IEEE Journal of*, v. 15, n. 3, p. 167–178, Jul 1990. ISSN 0364-9059. Cited on page 22.
- 30 JOYEUX, S.; ALBIEZ, J. Robot development: from components to systems. In: *6th National Conference on Control Architectures of Robots*. [S.l.: s.n.], 2011. p. 15–p. Cited on page 25.
- 31 FOOTE, T.; PURVIS, M. Rep 103—standard units of measure and coordinate conventions. *ROS. org/ Powering the world's robots*, 2014. Cited on page 27.
- 32 FOSSEN, T. I. *Guidance and control of ocean vehicles*. [S.l.]: John Wiley & Sons Inc, 1994. Cited 4 times on pages 27, 28, 29, and 30.
- 33 ANDRLE, M. S.; CRASSIDIS, J. L. Geometric integration of quaternions. *Journal of Guidance, Control, and Dynamics*, American Institute of Aeronautics and Astronautics, v. 36, n. 6, p. 1762–1767, 2013. Cited on page 28.
- 34 SCHWAB, A. L. Quaternions, finite rotation and euler parameters. *Cornell University Notes, Ithaca NY*, 2002. Cited on page 28.
- 35 IMLAY, F. H. *The complete expressions for added mass of a rigid body moving in an ideal fluid*. [S.l.], 1961. Cited on page 29.
- 36 MARTIN, S. C.; MARTIN, C. S. C. Advances in six-degree-of-freedom dynamics and control of underwater vehicles by. Citeseer, 2008. Cited on page 29.
- 37 MARTIN, S.; WHITCOMB, L. Preliminary experiments in comparative experimental identification of six degree-of-freedom coupled dynamic plant models for underwater robot vehicles. In: *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. [S.l.: s.n.], 2013. p. 2962–2969. ISSN 1050-4729. Cited 3 times on pages 30, 36, and 43.
- 38 GIBSON, S. B. et al. A comparison of hydrodynamic damping models using least-squares and adaptive identifier methods for autonomous underwater vehicles. In: *IEEE. OCEANS'15 MTS/IEEE Washington*. [S.l.], 2015. p. 1–7. Cited 2 times on pages 30 and 43.

- 39 SMALLWOOD, D. A. Advances in dynamical modeling and control of underwater robotic vehicles. The Johns Hopkins University, 2003. Cited 2 times on pages 30 and 47.
- 40 BRITTO, J. et al. Model identification of an unmanned underwater vehicle via an adaptive technique and artificial fiducial markers. In: IEEE. *OCEANS'15 MTS/IEEE Washington*. [S.l.], 2015. p. 1–6. Cited 4 times on pages 31, 36, 38, and 43.
- 41 WATANABE, T. et al. The rock-gazebo integration and a real-time auv simulation. In: IEEE. *Robotics Symposium (LARS) and 2015 3rd Brazilian Symposium on Robotics (LARS-SBR), 2015 12th Latin American*. [S.l.], 2015. p. 132–138. Cited 4 times on pages 33, 34, 35, and 38.
- 42 SMITH, R. et al. Open dynamics engine. 2005. Cited 2 times on pages 33 and 35.
- 43 JOHANSEN, T. A.; FOSSEN, T. I.; VIK, B. Hardware-in-the-loop testing of dp systems. In: *Dynamic positioning conference*. [S.l.: s.n.], 2005. Cited on page 33.
- 44 SKJETNE, R.; EGELAND, O. Hardware-in-the-loop simulation for testing of dp vessels. In: *IEEE Electric Ship Technologies Symposium (ESTS), Philadelphia*. [S.l.: s.n.], 2005. Cited on page 33.
- 45 PEREZ, T. et al. An overview of the marine systems simulator (mss): A simulink® toolbox for marine control systems. *Modeling, identification and Control*, Norwegian Society of Automatic Control, v. 27, n. 4, p. 259–275, 2006. Cited on page 33.
- 46 TOSIK, T.; MAEHLE, E. Mars: A simulation environment for marine robotics. In: IEEE. *Oceans-St. John's, 2014*. [S.l.], 2014. p. 1–7. Cited 2 times on pages 34 and 35.
- 47 ANITESCU, M.; POTRA, F. A. Formulating dynamic multi-rigid-body contact problems with friction as solvable linear complementarity problems. *Nonlinear Dynamics*, Springer, v. 14, n. 3, p. 231–247, 1997. Cited on page 35.
- 48 LEE, C.-H.; NEWMAN, J. N. Wamit user manual. *WAMIT, Inc*, 2006. Cited on page 42.
- 49 CD-ADAPCO, U. *Guide: Star-ccm+*. [S.l.]: Version, 2009. Cited on page 42.
- 50 YANG, R. et al. Modeling of a complex-shaped underwater vehicle for robust control scheme. *Journal of Intelligent & Robotic Systems*, Springer, v. 80, n. 3-4, p. 491–506, 2015. Cited on page 42.
- 51 ENG, Y.-H.; CHIN, C.-S.; LAU, M. W.-S. Added mass computation for control of an open-frame remotely-operated vehicle: Application using wamit and matlab. *Journal of Marine Science and Technology*, v. 22, n. 4, p. 405–416, 2014. Cited on page 42.
- 52 ROSS, A.; FOSSEN, T. I.; JOHANSEN, T. A. Identification of underwater vehicle hydrodynamic coefficients using free decay tests. *IFAC Proceedings Volumes*, Elsevier, v. 37, n. 10, p. 363–368, 2004. Cited on page 42.
- 53 ENG, Y. et al. Identification of the hydrodynamics coefficients of an underwater vehicle using free decay pendulum motion. In: *International MultiConference of Engineers and Computer Scientists*. [S.l.: s.n.], 2008. v. 2, p. 423–430. Cited on page 42.

- 54 CHAN, W. L.; KANG, T. Simultaneous determination of drag coefficient and added mass. *IEEE Journal of Oceanic Engineering*, IEEE, v. 36, n. 3, p. 422–430, 2011. Cited on page 42.
- 55 ALESSANDRI, A. et al. Application of ls and ekf techniques to the identification of underwater vehicles. In: *Control Applications, 1998. Proceedings of the 1998 IEEE International Conference on*. [S.l.: s.n.], 1998. v. 2, p. 1084–1088 vol.2. Cited on page 43.
- 56 PETRICH, J.; NEU, W. L.; STILWELL, D. J. Identification of a simplified auv pitch axis model for control design: Theory and experiments. In: IEEE. *OCEANS 2007*. [S.l.], 2007. p. 1–7. Cited on page 43.
- 57 NATARAJAN, S.; GAUDIG, C.; HILDEBRANDT, M. Offline experimental parameter identification using on-board sensors for an autonomous underwater vehicle. In: *Oceans, 2012*. [S.l.: s.n.], 2012. p. 1–8. Cited on page 43.
- 58 MISKOVIC, N.; VUKIC, Z.; BARISIC, M. Identification of coupled mathematical models for underwater vehicles. In: *OCEANS 2007 - Europe*. [S.l.: s.n.], 2007. p. 1–6. Cited on page 43.
- 59 VEN, P. W. V. D. et al. Neural network augmented identification of underwater vehicle models. *Control Engineering Practice*, Elsevier, v. 15, n. 6, p. 715–725, 2007. Cited on page 43.
- 60 SAYYAADI, H.; URA, T. Multi input-multi output system identification of auv systems by neural network. In: IEEE. *OCEANS'99 MTS/IEEE. Riding the Crest into the 21st Century*. [S.l.], 1999. v. 1, p. 201–208. Cited on page 43.
- 61 WEHBE, B.; HILDEBRANDT, M.; KIRCHNER, F. Experimental evaluation of various machine learning regression methods for model identification of autonomous underwater vehicles. In: IEEE. *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. [S.l.], 2017. p. 4885–4890. Cited on page 43.
- 62 PANG, Y. et al. On-line modeling of auv's maneuvering motion based on sliding window svm. In: ATLANTIS PRESS. *The International Conference on Artificial Intelligence and Software Engineering (ICAISE 2013)*. [S.l.], 2013. Cited on page 43.
- 63 KARRAS, G.; LOIZOU, S.; KYRIAKOPOULOS, K. On-line state and parameter estimation of an under-actuated underwater vehicle using a modified dual unscented kalman filter. In: *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. [S.l.: s.n.], 2010. p. 4868–4873. ISSN 2153-0858. Cited on page 43.
- 64 PINHEIRO, A. R. M.; BARROS, E. A. D. Identification of hydrodynamic derivatives from auv "pirajuba" using extended kalman filter. In: IEEE. *Computing and Automation for Offshore Shipbuilding (NAVCOMP), 2013 Symposium on*. [S.l.], 2013. p. 11–16. Cited on page 43.
- 65 TIANO, A. et al. Observer kalman filter identification of an autonomous underwater vehicle. *Control engineering practice*, Elsevier, v. 15, n. 6, p. 727–739, 2007. Cited on page 43.
- 66 SABET, M. T. et al. Identification of an autonomous underwater vehicle hydrodynamic model using the extended, cubature, and transformed unscented kalman filter. *IEEE Journal of Oceanic Engineering*, IEEE, 2017. Cited on page 43.

- 67 ENG, Y. H. et al. Online system identification of an autonomous underwater vehicle via in-field experiments. *IEEE Journal of Oceanic Engineering*, IEEE, v. 41, n. 1, p. 5–17, 2016. Cited on page 43.
- 68 PARK, C.-W.; CHO, Y.-W. Ts model based indirect adaptive fuzzy control using online parameter estimation. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, IEEE, v. 34, n. 6, p. 2293–2302, 2004. Cited on page 43.
- 69 KINSEY, J. C.; WHITCOMB, L. L. Adaptive identification on the group of rigid-body rotations and its application to underwater vehicle navigation. *IEEE Transactions on Robotics*, IEEE, v. 23, n. 1, p. 124–136, 2007. Cited on page 43.
- 70 OVASKA, S. J.; VALIWIITA, S. Angular acceleration measurement: A review. *IEEE transactions on Instrumentation and Measurement*, IEEE, v. 47, n. 5, p. 1211–1217, 1998. Cited on page 43.
- 71 ÅSTRÖM, K. J.; WITTENMARK, B. *Adaptive control*. [S.l.]: Courier Corporation, 2013. Cited on page 44.
- 72 SLOTINE, J.-J. E.; LI, W. et al. *Applied nonlinear control*. [S.l.]: Prentice hall Englewood Cliffs, NJ, 1991. v. 199. Cited 3 times on pages 44, 46, and 47.
- 73 DFKI GmbH. *Maritime Exploration Hall*. 2017. Available from: <<https://robotik.dfki-bremen.de/en/research/research-facilities/maritime-exploration-hall.html>>. Accessed: 2018-01-30. Cited on page 47.
- 74 MCFARLAND, C. *Adaptive identification and control for underwater vehicles: Theory and experimental evaluations*. Tese (Doutorado) — Ph. D. dissertation, The Johns Hopkins University, Baltimore, MD USA, 2013. Cited on page 47.
- 75 PEREIRA, M. A. G.; LESSA, G. C. Varying patterns of water circulation in canal de cotegipe, baía de todos os santos. *Revista Brasileira de Geofísica*, SciELO Brasil, v. 27, n. 1, p. 103–119, 2009. Cited on page 51.
- 76 BRITTO, J. et al. Improvements in dynamics simulation for underwater vehicles deployed in gazebo. In: IEEE. *OCEANS–Anchorage, 2017*. [S.l.], 2017. p. 1–6. Cited on page 66.
- 77 SAVITZKY, A.; GOLAY, M. J. Smoothing and differentiation of data by simplified least squares procedures. *Analytical chemistry*, ACS Publications, v. 36, n. 8, p. 1627–1639, 1964. Cited on page 68.
- 78 SCHAFER, R. W. What is a savitzky-golay filter?[lecture notes]. *IEEE Signal processing magazine*, IEEE, v. 28, n. 4, p. 111–117, 2011. Cited on page 68.
- 79 GORRY, P. A. General least-squares smoothing and differentiation by the convolution (savitzky-golay) method. *Analytical Chemistry*, ACS Publications, v. 62, n. 6, p. 570–573, 1990. Cited 2 times on pages 68 and 69.
- 80 THORNLEY, D. J. Novel anisotropic multidimensional convolutional filters for derivative estimation and reconstruction. In: IEEE. *Signal Processing and Communications, 2007. ICSPC 2007. IEEE International Conference on*. [S.l.], 2007. p. 253–256. Cited 2 times on pages 68 and 69.

- 81 KORTELAINEN, J. et al. Eeg frequency progression during induction of anesthesia: from start of infusion to onset of burst suppression pattern. In: IEEE. *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*. [S.l.], 2007. p. 1570–1573. Cited on page 68.
- 82 HARGITTAI, S. Savitzky-golay least-squares polynomial filters in ecg signal processing. In: IEEE. *Computers in Cardiology, 2005*. [S.l.], 2005. p. 763–766. Cited on page 68.
- 83 PAN, B. et al. Full-field strain measurement using a two-dimensional savitzky-golay digital differentiator in digital image correlation. *Optical Engineering*, International Society for Optics and Photonics, v. 46, n. 3, p. 033601, 2007. Cited on page 68.
- 84 SUMAN, S.; JHA, R. K. A new technique for image enhancement using digital fractional-order savitzky–golay differentiator. *Multidimensional Systems and Signal Processing*, Springer, v. 28, n. 2, p. 709–733, 2017. Cited on page 68.
- 85 LUYPAERT, J.; ZHANG, M.; MASSART, D. Feasibility study for the use of near infrared spectroscopy in the qualitative and quantitative analysis of green tea, *camellia sinensis* (l.). *Analytica Chimica Acta*, Elsevier, v. 478, n. 2, p. 303–312, 2003. Cited on page 68.