N° Ordre........../FT/UMBB/2021

PEOPLE’S DEMOCRATIC REPUBLIC OF ALGERIA

MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH

UNIVERSITY M’HAMED BOUGARA OF BOUMERDES

Faculty of technology

Solid Mechanics and Systems Laboratory - LMSS

Doctorate thesis

Presented by:

Touzout Walid

With a view to obtaining the DOCTORATE degree in:

Field: Mechanical Engineering

Option: Mechatronic

TITLE:

Energy consumption modelling of marine drones and the integration of the model into ROS-based simulation.

<u>In front of the jury composed of:</u>

Mr. Adjerid	Smail	Professor	UMBB	Chairman
Mr. Benazzouz	Djamel	Professor	UMBB	Thesis director
Mr. Benmoussa	Yahia	MCA	UMBB	Thesis co-director
Mr. Bentarzi	Hamid	Professor	UMBB	Examiner
Mr. Kara	Redouane	Professor	UMMTO	Examiner
Mr. Rahmoune	Chemseddine	MCA	UMBB	Invited
Mr. Diguet	Jean Philippe	HDR	UBS-Lorient	Invited

Academic year: 2020/2021

Dedications

This thesis is dedicated to my lovely parents for their endless support and encouragement and to my dear brothers. I further extend my dedication to all members of Touzout's family.

I dedicate this work also without exception to all my friends, colleagues, and my teachers.

Last and not least, I dedicate my PhD to everyone who has taught, encouraged, and advised me during all my studies.

Acknowledgments

First and foremost, all praise and thanks giving to Allah the most powerful and most merciful who gave me the ability and patience to accomplish the work presented.

This thesis was carried out in the laboratory of Solid Mechanics and Systems of the University of Boumerdes. I would like to express my sincere thanks to my thesis supervisor Pr Benazzouz Djamel who supervised me and shared with me his brilliant insights throughout this thesis and offered to me a very pleasant research environment.

One of the most rewarding experiences of my PhD was having the chance to work with my co-supervisor, Dr Benmoussa Yahia. I am very grateful for his involvement in my research work, his wealth of knowledge in the field was extremely helpful. Dr Benmoussa Yahia was an excellent advisor, always available to discuss and support all technical issues and perspectives.

My deep gratitude is owed to Dr Jean Phillipe Diguet from the University of Bretagne-Sud UBS, Lorient, France for the confidence he placed in me by giving to me the opportunity to work in the Lab-STICC laboratory and for sharing his valuable experience.

I would like to thank Dr Erwan Moreac from UBS University who accompanied me during my stay in the Lab-STICC laboratory where he gave me full support and shared his valuable experience and technical skills with me.

I would like to express my gratitude to Prof Adjerid Smail, for the honour he conferred on me by chairing the jury of this thesis, I would also like to express my special gratitude to the rest of my thesis committee: Prof Kara Redouane, Prof Bentarzi Hamid, and Dr Rahmoune Chemessedine for being the members of my thesis jury and for the valuable time they spent evaluating the work presented.

I would like to express my gratitude to all the members of the Solid Mechanics and Systems Laboratory for their support and encouragement.

Abstract

The Unmanned Surface Vehicles (USVs) are promising solutions for various marine applications such as: maritime navigation, rescue, environmental control, military missions, oceanic maps production, etc. The main advantage of USVs is the ability to execute their functionalities in environments where humans are not able to intervene safely, in addition to their cost and continuous activity.

Generally, USVs operate in difficult environmental conditions requiring precision, reliability, and autonomy. To meet these critical requirements, the scientific community is increasingly focusing its research in the USV's field and their applications. Accordingly, one of the most difficult issues to be resolved in this field is the autonomy and energy limitation problems. Estimating and managing the power consumption of USVs is an important issue to deal with energy minimization techniques such as trajectory planning, task scheduling and optimal design of controllers. In this thesis, we present the energy consumption parameter of USVs into Robot Operating System (ROS) - based simulation through the following contributions:

- An analytical model of the energy consumption of differential drive Unmanned Surface Vehicles is developed based on a three-degrees-of-freedom dynamic model of surface vessels.
- A reverse engineering approach is proposed allowing the identification of the developed dynamic model's coefficients and parameters based on a set of scenarios run within the simulation environment presented in [1]. The identified model is used in the development of the consumption model of surface vehicles.
- The simulator engine is enriched with power modelling and simulation tools,

so that the power consumed by the USV is instantaneously calculated, processed, and returned; thus, the energy required to accomplish a given predefined scenario is available as a new simulation result.

Keywords: Autonomous vehicles, marine drones, unmanned surface vehicles, differential drive boats, energy consumption, modelling and simulation, robotic simulation environment, Robot Operating System (ROS), Gazebo software, USV simulator.

Résumé

Le développement des véhicules marins de surface sans pilote (Unmanned Surface Vehicles USV) est une solution prometteuse pour diverses applications marines telles que : la navigation maritime, le sauvetage, le contrôle environnemental, les missions militaires, la production de cartes océaniques, etc. Le principal avantage des USVs est leur capacité à exécuter leurs fonctionnalités dans des environnements où l'homme ne peut pas intervenir en toute sécurité, en plus de leur coût et de leur activité continue.

En fait, les USVs opèrent dans des conditions environnementales difficiles qui exigent : La précision, la fiabilité et l'autonomie. Pour répondre à ces exigences critiques, la communauté scientifique concentre de plus en plus ses recherches dans le domaine des USV et de leurs applications. En conséquence, l'un des problématiques les plus difficiles à résoudre dans ce domaine est celui de l'autonomie et de la limitation de la consommation d'énergie. L'estimation et la gestion de la consommation d'énergie des USV est une question très importante à traiter avec des techniques de minimisation de l'énergie telles que : la planification de la trajectoire, l'ordonnancement des tâches, la conception optimale des régulateurs et contrôleurs, etc. Dans cette thèse, nous introduisons le paramètre de consommation d'énergie des USVs dans la simulation basée sur le système d'exploitation de robot (Robot Operating System ROS) à travers les contributions suivantes :

– Un modèle analytique de la consommation d'énergie des véhicules de surface sans pilote à entraînement différentiel est développé sur la base d'un modèle dynamique à trois degrés de liberté des navires de surface.

- Une approche de « reverse engineering » est proposée, permettant l'identification des coefficients et des paramètres du modèle dynamique développé sur la base d'un ensemble de scénarios exécutés dans l'environnement de simulation présenté dans [1]. Le modèle identifié est exploité dans le développement du modèle de consommation des drones marins.
- Le logiciel de simulation est enrichi avec des outils de modélisation et de simulation de l'énergie, de sorte que la puissance absorbée par le drones marin (USV) est instantanément calculée, traitée et restituée ; ainsi, l'énergie requise pour accomplir un scénario prédéfini est disponible sous forme de nouveau résultat de simulation.

Mots-clé : Véhicules autonomes, drones marins, véhicules de surface sans pilote, bateaux à entraînement différentiel, consommation d'énergie, modélisation et simulation, environnement de simulation robotique, Robot Operating System (ROS), logiciel Gazebo, simulateur USV.

الملخص

تعتبر المركبات المائية السطحية ذات تحكم ذاتي (بدون طيار) (Unmanned Surface Vehicles USV) حلولًا واعدة للعديد من التطبيقات البحرية مثل: الملاحة البحرية ، الإنقاذ ، المراقبة البيئية ، المهمات العسكرية ، إنتاج خرائط المحيطات ، وما إلى ذلك. الميزة الرئيسية لذه المركبات هي قدرتها على العمل في البيئات التي لا يستطيع البشر التدخل فيها بأمان إضافةً إلى تكلفتها ونشاطها المستمر.

تعمل المركبات السطحية الذاتية USV أساسا في ظروف بيئية صعبة تتطلب الدقة ، الموثوقية ، والاستقلالية. لتلبية هذه المتطلبات الحرجة ، يركز المجتمع العلمي بشكل متزايد أبحاثه في مجال هذا الصنف من المركبات وتطبيقاتها سواءا كانت طائرات بدون طيار ، مركبات تحت الماء الذاتية ، أو المركبات السطحية الذاتية USVs. وفقًا لذلك ، فإن إحدى أصعب المشكلات التي يجب حلها في هذا المجال هي مشكلات الاستقلالية و كمية الطاقة المحدودة. يعد تقدير وإدارة استهلاك الطاقة من طرف USV مسألة مهمة للتعامل مع تقنيات تقليل و ترشيد الطاقة مثل: تخطيط المسار ، جدولة المهام ، التصميم الأمثل لوحدات التحكم ، إلخ.

في هذه الأطروحة ، نقدم مَعْلَمَة الاستهلاك في محاكاة المركبات السطحية من خلال ثلاث مساهمات:

1) تطوير نموذج رياضي لاستهلاك الطاقة للمركبات السطحية على أساس نموذج ديناميكي ثلاثي درجات الحرية للسفن المائية.

2) اقتراح نهج الهندسة العكسية ، مما يسمح بتحديد معاملات النموذج الديناميكي المستخدم سابقًا ، بناءً على مجموعة من السيناريوهات التي تم تشغيلها داخل بيئة المحاكاة المعروضة في [1].

3) إثراء محرك المحاكاة بأدوات النمذجة الطاقوية ، بحيث يتم حساب الاستطاعة اللحظية الممتصة من طرف مركبة سطحية USV و عرضها بيانيا بدلاية الزمن وبالتالي ، فإن الطاقة المطلوبة لإكمال أي سيناريو محدد مسبقًا متاحة كنتيجة محاكاة جديدة.

الكلمات المفتاحية

الروبوتات ذات التحكم الذاتي ، المركبات السطحية الذاتية USV، استهلاك الطاقة ، النمذجة ، المحاكاة ، برامج محاكاة الروبوتات المحمولة ، نظام تشغيل الروبوت ROS .

Table of Contents

List of Figures

List of Tables

Abbreviations

AUV	Autonomous Underwater Vehicle
DDR	Differential Drive Robot
DOF	Degrees Of freedom
FWMMR	Four Wheeled Mecanum Mobile robot
GNC	Guidance, Navigation, and Control
GPS	Global Positioning System
GUI	Graphical User Interface
HDF	Hierarchical Data Format
MMG	Mathematical Model Group
PWM	Pulse Width Modulation
ROS	Robot Operating System
ROV	Remotely Operating Vehicle
Rviz	Robot visualization
SSMO	Skid-Steer Mobile Robot
TOMR	Three-wheeled omnidirectional Mobile Robot
UAV	Unmanned Aerial Vehicle
UGV	Unmanned Ground Vehicle
UMV	Unmanned Maritime Vehicle
URDF	Unified Robot Description Format
URV	Unmanned Robotic Vehicle
USV	Unmanned Surface Vehicle
USVsim	USV simulator
UUV	Unmanned Underwater Vehicle
UWsim	Under Water simulator

VMRC	Virtual Maritime RobotX Competition
XML	Extensible Markup Language

Chapter 1

Introduction

1.1 Context

The Unmanned Maritime Vehicles (UMV) are boats that operate on the water surface without a crew. They can operate autonomously or can be remotely controlled. Basically, UMVs are divided into two types: Unmanned Underwater Vehicles (UUV) and Unmanned Surface Vehicles (USV). UUV generally refers to those vehicles without human crew, with either automated navigation, such as Autonomous Underwater Vehicles (AUV) or Remotely Operated Vehicles (ROVs). AUV are usually more smaller than their USV counter parts [2]. ROVs have the main property of having a cable, called umbilical cord, which connects the vehicle with a surface boat, allowing in this way communication, power transmission, and interchange of data or control commands [3, 4].

On the other hand, the USVs, which are targeted in this study, are a type of boat surfing the water surface or semi-submerged. Figure 1.1 describe the components of a typical USV :

1. Communication system
2. Sensors
3. Guidance, Navigation, and Control (GNC) system
4. Ground station

5. Propulsion and power system

6. Hull (Ex. Kayaka, Catamaran or Trimaran)

The sensors are used to collect various environmental data which can be processed locally or fed to the ground station. Besides, the GNC is used to control the USV. It may operate autonomously or receive commands from the ground station. All these features make USVs a promising solution for different marine applications such as : port navigation, rescue, environment control, oceanic maps generation, Oil and Gas Industry, [4, 5], fluvial environments [2], military development [6], detecting contaminants. [7].

The main advantages of this technology over manned vehicles are that the USV is not constrained due to restrictions imposed by a human crew member, such as temperature, space, or environment disturbances, as well as their ability to evolve in environments where humans are not able to intervene safely in addition to their cost and continuous activity [6].

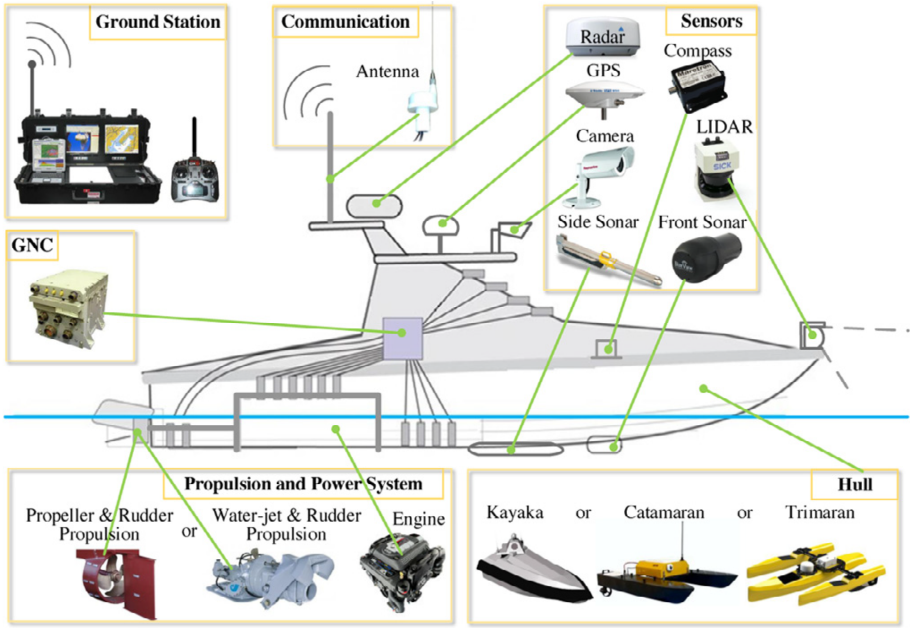


Figure 1.1: Typical USV architecture

1.2 Problem statement

1.2.1 USV's energy autonomy

USVs operate in difficult environment conditions needing precision, reliability, and autonomy and requiring advanced features, peripherals, and accessories to handle these complicated tasks. To provide solutions to these critical requirements, scientific community is more and more focusing their research in the USVs field and their applications [8]. One of the most challenging issues which should be considered when planning any kind of USV's mission is the autonomy problem. Actually, the ability of USVs to perform their missions is limited by the capacity of their batteries. However, this issue is more critical in case of mission operating in an environment where any battery discharges may have a considerable impact on safety and/or cost [9].

1.2.2 Source of energy consumption

The power consumption tree of USVs is given in Figure 1.2. It is illustrated that the total power is absorbed by: the on-board computer, sensors, control systems, thruster's losses, and mainly: thrusters' useful power converted to the mechanical energy generating the drones' movements. Basically, the power absorbed by the on-board computer, losses, and electrical components (sensors, control system, etc.) is a small quantity compared to that of the thruster useful power consumption [10]; thus, they can be approximated by a constant value or more accurate model. As a result, the main power consumption is due to thrusters (80-90%) transferred to the mechanical power which should be modelled and estimated according to the USV speed, movement, and environment conditions. [11]

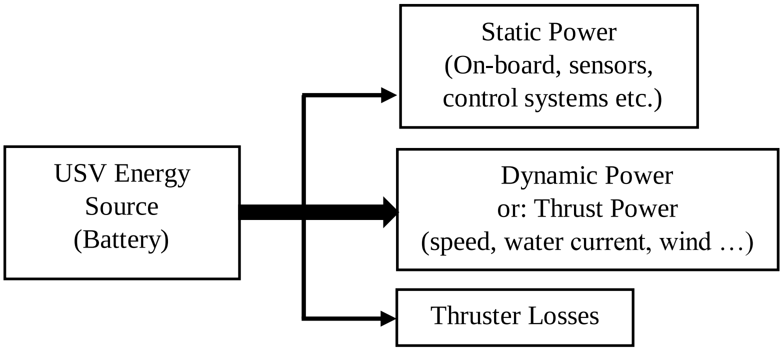

Static Power
(On-board, sensors, control systems etc.)
USV Energy Source (Battery)
Dynamic Power or: Thrust Power (speed, water current, wind …)
Thruster Losses

Figure 1.2: Energy consumption of the USV.

1.2.3 Saving the energy consummation

Actually, the USVs autonomy depends on their battery capacity and also on their environmental parameters such as wind and water current. Considering all the parameters makes the consumption optimization one of the most important challenges for the research community working on the USVs field. Thus, the estimation of the USV's energy consumption depends on their motion planning and environment disturbances.

Basically, power reduction techniques of unmanned vehicles may be applied at all levels of the system design hierarchy. These levels include: physical level (equipment's and circuits architecture), system level, and application level. A brief description of each level is given followed by some specific applications.

The power reduction in the physical level depends on the types of propulsion/power system and electronic devices used in a system and their electronic architecture like different ICs' types (CMOS or TTLs), CPUs, types of conductors, etc [12]. Many techniques for power reduction are available at this level such as optimal controller design [13, 14, 15]. In this level, the resultant energy consumption estimation can be achieved by measuring instantaneously the real power absorbed by the vehicle using on-board sensors (mainly current and voltage sensors) [16].

Moreover, the power reduction in the system level can be achieved, for instance, by generating optimal tasks scheduling. This approach can be supervised by an operating system used to manage all common resources within a system including power resource. When all processes are properly scheduled, power reduction could be achieved without degrading the vehicle's performances [17, 18]. As a consequence, these power reduction techniques can be used to only reduce the static power consumption of the unmanned vehicle; however, and as previously reported, the main consumption part of the USVs is due to their dynamics and environment disturbances. Thus, a much higher level of power minimization is required mainly the application level.

At the application level, several approaches have been proposed aiming to minimize the power consumption by using mainly optimal path planning generation [9, 19, 20]. Theses solutions may provide good results but their main drawback is the use of simple and non-realistic energy models which are not very representative of real-life USV consumption.

Therefore, another interesting approach consists to use of simulation environment to test and to validate USVs properties at early phase of mission planning which allows saving cost and time [1]. However, the existing simulation environment do not consider the energy consumption parameters within their engines.

1.3 Contributions

The proposed solution in this thesis begins by developing a more realistic power model of the USVs based on their dynamic behaviours; thereafter, the model is integrated into one of the robust and the most relevant USV simulators that exist which is based on Robot Operating systems ROS platforms. Recently, many USV's simulation environments have been shared such as the one presented in [1], this latter is an open source, recent, and very robust simulator that simulates different USV types under realistic disturbances. However, none of the available simulators considers the energy consumption parameter in their engines.

In this thesis, we introduce the energy consumption parameter in USVs simulation. Our methodology is implemented on the open-source simulation environment presented in [1] and it can be generalized and exploited with the other USV simulation platforms. The used USV simulator simulates four virtual USV types (air boats, sail boats, differential drive boats, and ruder boat) in realistic environment under realistic disturbances. Actually, the energy consumption parameter introducing into the simulator has been done through the following steps: First, an analytical USVs energy consumption model is developed based on the three-degrees-of-freedom (3-DOF) dynamic model given in [21] by ignoring the wave effects. Then, we illustrate the identification approach proposed to define the constant coefficients of the used dynamic model based on a set of scenarios executed within the simulation environment based on a reverse engineering approach [22]. We apply this methodology particularly for Lutra-Prop boat, a representative differential drive USV available in the used simulation environment. Thereafter, the energy model obtained from the two previous steps is integrated into the simulator engine since the simulator is offered on an open source package [23]. Finally, we run different simulation scenarios and we analysed the obtained estimated energy consumption results and responses in a short time without any hardware requirement. In addition, the presented approach allows to evaluate the effect of the USV speed on its consumption as given later in this thesis.

1.4 Methodology

The methodology followed in our thesis starts as given in Figure 1.3 by introducing the USV dynamics and their equation of motions alongside with water current and wind speed disturbances in order to obtain the 3-DOF dynamic model. After that, the model parameters have been identified using simulator-based empirical study known as: Reverse engineering approach; thereafter the power model can be analytically deduced. Therefore, the energy consumption of USV can be represented as a function its speed and environment disturbances.

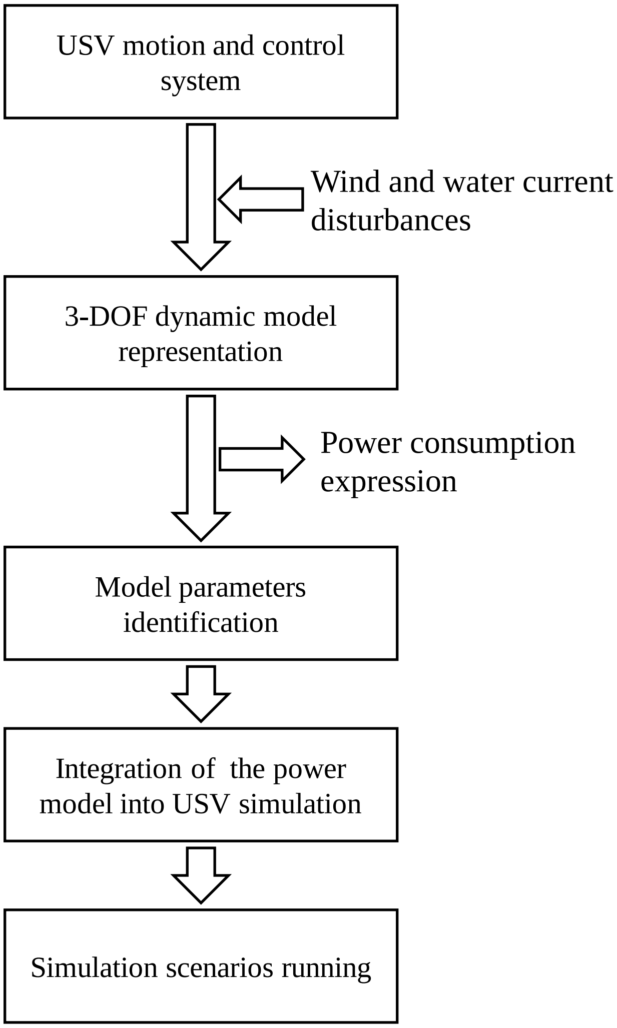

USV motion and control system
Wind and water current disturbances
3-DOF dynamic model representation
Power consumption expression
Model parameters identification
Integration of the power model into USV simulation
Simulation scenarios running

Figure 1.3: Methodology workflow

In order to show the effectiveness of the energy modeling and integration into the simulation environment approach, different scenarios were conducted with and without the presence of wind and water current disturbances. For each scenario, we recorded the surge velocity u(m/s) and the instantaneous dynamic power absorbed by the USV. Table 1.1 represents the description of seven different scenarios in terms of the vehicle's speed and environment disturbances.

Table 1.1: Verification scenarios

Scenario #	Description	Water current speed (m/s)	Water current direction	Wind speed (m/s)	Wind direction
1	USV moves from the starting point (240,95) to the target point (270,95) at maximum speed	0.0	0.0°	0.0	0.0°
2	USV moves from the starting point (240,95) to the target point (270,95) at maximum speed	0.40	0.0°	0.0	0.0°
3	USV moves from the starting point (240,95) to the target point (270,95) at maximum speed	0.40	45.0°	0.0	0.0°
4	USV moves from the starting point (270,95) to the target point (240,95) at maximum speed	0.40	45.0°	0.0	0.0°
5	USV moves from the starting point (240,95) to the target point (270,95) with maximum linear speed of 1 m/s	0.0	0.0°	0.0	0.0°
6	USV moves from the starting point (240,95) to the target point (270,95) with maximum linear speed of 1 m/s	0.30	45.0°	0.20	30.0°
7	USV moves from the starting point (270,95) to the target point (240,95) with maximum linear speed of 1 m/s	0.30	45.0°	0.20	30.0°

1.5 Thesis organization

The remaining chapters of this thesis are organized as follows:

- Chapter 2 involves the literature review from different perspectives. First, the energy consumption of mobile robots generally and marine drones especially have been discussed. Thereafter, we introduce a survey on the simulator-based applications by highlighting the most recent, relevant, and the most interesting related work.

- The next chapter (Chapter 3) includes an overview of the Robot Operating Systems(ROS)-based simulation tools. This chapter gives: overviews, highlights, descriptions, and surveys of the main ROS-based software used to achieve the thesis objectives such as: Gazebo software, USV simulator, Under Water UW simulator, and custom ROS processes known as ROS nodes.

- Chapter 4 involves the analytical approach of the development the power consumption model of marine drones as a function of their speed vector and environment disturbances (mainly: wind force and water current). The obtained model is given as a matrix form representation using the USV dynamic parameters and characteristics identified as given in the next chapter.

- Chapter 5 illustrates the model parameters identification of the Lutra-prop differential drive USV's model using a reverse-engineering approach from the used USV simulator software. The obtained parameters are used to complete the power model of the USV in addition to its dynamic representation.

- Chapter 6 demonstrates the integration of the obtained power model into its corresponding virtual package within the USV simulator by implementing additional processes to enrich the ROS-based software infrastructure since the provided USV simulator is a Linux-based open source software. In order to verify the proposed approach's results, seven different realistic scenarios are proposed in this chapter where the instantaneous power was recorded and plotted; thus, the total energy consumption is returned at the end of each scenario. Furthermore, we provide also in this chapter the variation representation of the effect of the USV's speed on its power consumption.

- At the end, the thesis' report is ended up with a general conclusion summarizing the presented work and contributions, as well as it gives an insight to further feasible works.

Chapter 2

Literature review

2.1 Introduction

USVs are always in competition with other manned or unmanned systems in terms of some specific applications. Therefor, the future progress of USVs depends on the development of full-autonomy, enabling USVs to work in any unstructured or unpredictable environment without any human supervision and with the minimum energy consumption control. The development of such an autonomy is very challenging, since it requires the development of effective and reliable USV systems, including reliable communication systems, suitable hull design, powerful GNC (Guidance, Navigation, and Control) strategies (see Figure 2.1), as well as optimal energy consumption systems design.

Despite strong demand for comprehensive reviews reporting, organizing, and comparing the large diversity of existing USV research field, we could involve, fortunately, in this chapter the most relevant work from different perspectives mainly: analytical modelling, autonomy problem, energy consumption, optimization approaches, simulation environments, etc.

In recent years, different USVs are studied by many universities and research institutes of ocean technologies around the world which have been applied in many

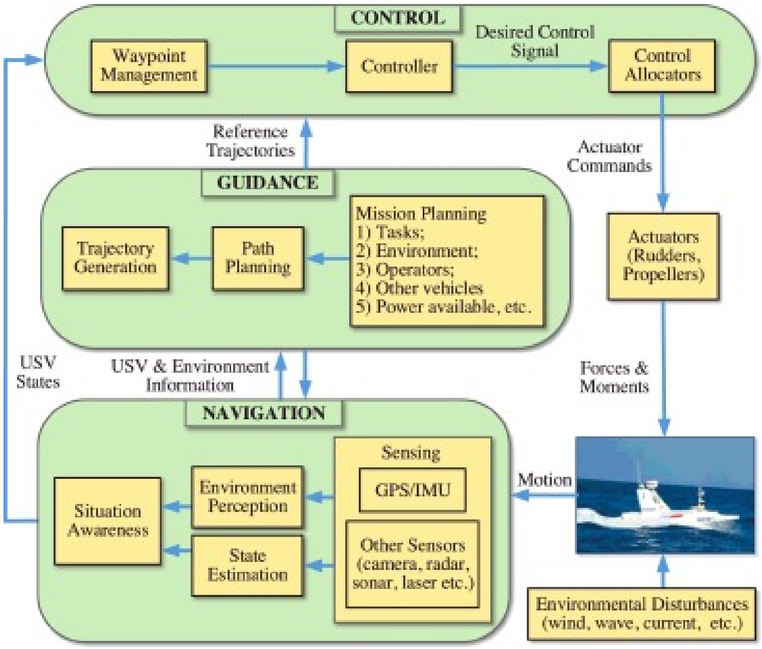


Figure 2.1: General structure of USV guidance, navigation, and control systems.

oceanic fields, such as bathemetry [24], environment monitoring [25], underwater acoustics [26], marine rescue [27] , goal tracking [28, 29], etc. Though related technologies for USVs are gradually mature, there are many challenges for USVs' control, such as appropriate model in complex environment, stable controller, obstacle avoidance, etc.

For effective energy estimation, motion study, design, and control of USVs, it is highly required to establish a mathematical and analytical models for their dynamics, kinematics, and kinetic performance [30, 27] which represent the basis for realizing intelligent control, maneuverability prediction, or energy cost estimation.

2.2 USV dynamics and modelling

Basically, there are two kinds of mathematical models for ship motion. One is the Abkowitz model [31], which is also called the global model. It considers the hull, propeller, and rudder as an integrated whole, and expands the hydrodynamic force acting on the hull into the Taylor series of each motion variable. The other is the so-called Mathematical Modeling Group (MMG) model [32], also known as the separable model. It was proposed by the Japanese MMG based on the Abkowitz model. The MMG model decomposes hydrodynamic forces into three parts: hull, propeller, and rudder, and takes the interaction between them into account. The Abkowitz model is mathematically more complete and rigorous, and therefore, widely used in the community. Based on the Abkowitz model, Fossen et al. [33, 21] put forward a method of modeling and control for marine craft, including ships, high-speed craft, semi submersibles and floating rigs. In our thesis, we used the Fossen's model describing the USV dynamics to generate the consumption model as illustrated in the upcoming chapters.

2.3 Energy consumption study of unmanned systems

Many recent works have been published aiming to model, to estimate, and to minimize the energy consumption of different mobile robots. The various energy loss components in differential drive robots are presented and well-defined in a complete energy model developed in [34]. The energy consumption of the mobile robot is modeled in [35] by considering three major factors: the sensor system, control system, and actuators. The relationship between the three systems is elaborated by formulas; thereafter, the model is utilized and experimentally tested in a four-wheeled Mecanum mobile robot. A power consumption modeling and analysis of the omnidirectional mobility of the three-wheeled omnidirectional mobile robot (TOMR) is proposed in [36]. When TOMRs navigate the target point at a given angles, the speed of each wheel changes dramatically; and therefore, the power consumption of that robot will also be greatly changed. Moreover, an approach is given in [37] which demonstrates the evaluation of Skid-Steer Mobile Robots (SSMR) power consumption based on slip parameters that are calculated as differential equations extracted from the equations of motion. The dynamic power model is validated and then implemented on two practical manufacturing applications in which the autonomous vehicle is climbing on steel surfaces with primary power consumption due to turning and overcoming gravity. A power model based on the dynamic parameters of the Differential Drive Robot (DDR) mobile robot and its motors is proposed in [38, 39]. The model can predict the consumed power accurately even if the robot carries different payloads or performs different accelerations. Moreover, the power consumption model for an omnidirectional movement of a Four-wheel Mecanum mobile Robots (FWMMR) is proposed in [15]. This energy model was mathematically implemented in MATLAB and validated by an experimental study.

As a result, most researchers consider the energy efficiency to be the key on autonomous robots' performances and autonomy which is of great significance for reducing the power consumption, motion control, and path planning of mobile robots since they are constrained by their batteries' capacity limitation. From the given literature review, it can be noticed that most of the related works were about designing power models for ground robots only to estimate their energy consumption.

2.4 Energy consumption study of USV

One of the contributions presented in this thesis is the development of the energy consumption model of an unmanned vehicle type which is the surface drone (or marine drone) called USVs. As previously reported, the consumed power of the USVs is that absorbed by the on-board computer, sensors, control systems, thruster's losses, and mainly thrusters' useful power generating the mechanical energy which depends on the boats shapes, dynamic behaviours, and environment conditions.

As previously reported, the power absorbed by the on-board computer, electrical loads, and electrical components (sensors, control system, etc.) is a small quantity compared with the thruster useful power consumption [10]; thus, it can be approximated by a constant value or a more accurate model. Therefore, the main power consumption is due to thrusters (80-90%) transferred to the mechanical power which should be modelled according to the USV dynamics, odometry, and environment conditions (see Figure 1.2).

The development of the consumption model of USV as function of its environment can be achieved using its dynamic model expressing the relationship among the velocity and a set of forces applied on the boat's body as given later in chapter 4. Thus, many relevant works have been taken into consideration to analytically develop the power model of the USVs and thereafter to identify its parameters: Muske K. R. et al [13] have presented a solution to identify the parameters of a 3-DOF non-linear dynamic surface vehicle model and validated their approach by carrying out on an experimental model. In addition, Li. C et al [40] have presented a 3-DOF dynamic model of a rudderless double thruster USV and the identified parameters of a 1.5m long, 50Kg USV through a system identification approach, the accuracy of their modelling and identification approaches has been verified by an experimental testing. Niu. H et al [9] have studied the energy efficient path planning algorithms for USV and studied in [41] the energy-based efficient path planning for USV in spatially and temporally variant environment; however, the considered cost function is a simple and non accurate 2-DOF power consumption model which is used to calculate the consumption weight of a given path. Mu. D et al [42] have presented a model and an identification approach of the propulsion vector of a USV based on its 3-DOF dynamics and response model, and compared the simulated turning and zigzag tests with a physical 7.02$m \times$2.6m USV. Jin. J et al [27] have designed a non-linear controller for the heading and the velocity of a USV based on 3-DOF model and identified its parameters. Sonnenburg C. R et al [43] have also described in their research paper the USV modelling, identification approaches, and control systems design. Wirtensohn. S et al [44] have presented a model of a twin hull-based USV and identified the model parameters via a weighted least square approach.

As given in the literature review, It can be summarized that energy estimation, modelling, and identification approaches may require much complicated tasks and eventually much hardware set-up and much time to be spent; furthermore, the obtained data cannot not be very precise due to the presence of disturbances especially in case of many scenarios or realistic ones. This problematic can be solved, as given in our thesis, by establishing a suitable and more realistic power model of USVs consumption based on their accurate 3-DOF dynamic behaviours representations; thereafter, the integration of the obtained model after identifying its parameters into a robust simulator engine that simulates the USV performances in realistic way under realistic disturbances is important to test, to verify, or to estimate the USV consumption.

2.5 Simulator-based applications review

Realistic simulation and prototyping of autonomous vehicles generally, or surface drones especially, play a significant role to reduce the hardware set-up and the amount of time spent in developing a challenging mobile robot application such as the estimation and the minimization of the energy consumption. Moreover, simulation and estimation tools allow designers and researchers to focus on the interesting parts of their applications to achieve better results. However, before simulating a given robotic tasks or scenarios, it is mandatory to choose a suitable simulation environment from a given perspective since different simulators offer different performances. In addition to the above research works: Kramer. J et al [45] have presented a study that addresses autonomous mobile Robot Development Environments (RDE) by comparing a set of open-source freely available RDEs from different points of view. ZlajpahL et al [46] have given an overview of the simulation in robotics field by showing how simulation makes things easier, as well as they have presented some advantages and drawbacks of simulation in robotics. Pitonakova. L et al [47] provide comparison study between the most used robot simulators involving: Gazebo, V-REP, ARGoS, etc. Consequently, many open source software platforms are available in order to simulate robotic behaviours in realistic environment such as Gazebo and ROS which stands for Robot Operating System. Gazebo software was developed in 2002 at the University of Southern California [48]. The idea was about creating a high-fidelity simulation environment that provides the ability to simulate robots in outdoor and indoor environments under different operating conditions; hence, Gazebo is a 3D robot simulator while ROS serves as the robot's interface.

Fortunately, ROS and Gazebo-based robot simulators are free and open source platforms (Linux-based), which help researchers to customize the simulator engines to simulate their applications especially those for unmanned vehicles. Among the most robust and high-fidelity Gazebo-based simulators that exist we find the Unmanned Ground Vehicle UGV simulator [49], UUV simulator [50], Unmanned Aerial Vehicle UAV simulator [51], and finally the USV simulator [1]. The latter is our case of study simulator platform since it is a recent and a very robust ROS and Gazebo-based package. The software simulates realistically the effect of waves, wind, buoyancy, water current, underwater behaviours, fluid dynamics, etc. It is provided with four different USV types (differential drive boat, air boat, sail boat, and a ruder boat) that can be controlled under different environment conditions and disturbances in different modes (time-varying or surface-varying) using Gazebo plugins that apply analytically represented forces and torques on the boat shape such as the hydrodynamic force, the hydro-static force, the wind force, the wave force, and the propulsion forces. As a consequence, and based on the virtual forces applied on the boat within the simulator, the virtual boats can behave in the virtual space as close as possible to the physical one; however, the energy parameter is not introduced within that simulator package which motivates us to enrich and to customize its engine with power management and estimation tools by integrating an analytical power consumption model into its package. Since the simulator is a ROS-based software, the energy parameter integration approach is done by implementing additional ROS nodes interfaced with the USV simulator files.

Chapter 3

ROS-based simulation tools

Mobile robotic applications are increasingly entering the real and complex world of humans in a way that requires a high degree of interaction and co-operation between humans and robots. Complex simulation models, expensive hardware configuration, and a highly controlled environment are often required during the various stages of robot development. Robot developers and researchers need a more flexible approach to conduct experiments and better understand how robots perceive the world. Mixed reality (MR) presents a world where real and virtual elements co-exist. By merging the real and the virtual in the creation of a MR simulation environment, it is possible to better understand the robot's behaviour and to create cheaper and safer test scenarios by making interactions between physical and virtual objects possible.

Actually, Robot developers are free to introduce virtual objects into a MR simulation environment to evaluate their systems and to obtain consistent visual feedback and realistic simulation results such as the energy consumption and minimization especially when dealing with unmanned vehicles operating in complex and critical environments such as the USVs. We illustrate in this chapter the most recent and robust USV simulator which is MR simulation tool built on the basis of ROS and Gazebo 3D robot simulator.

3.1 Introduction to ROS

ROS is an open source robotic software system that can be used without licensing fees by universities, government agencies, or any commercial companies. The advantages of an open source software is that the source code for the system is available and can be modified according to a user's needs. Therefore, the software can be improved and modules can be added by users or any research team.

ROS (or Robot Operating System) is a meta operating system that can perform many functions of an operating system; indeed, it requires a computer's operating system such as Linux. The main purposes of using ROS is to provide communication between the user, the computer's operating system, and external equipment. This equipment can include: sensors, cameras, or any robot peripheral. As with any operating system, the benefit of ROS is the hardware abstraction and its ability to control a robot without the user having to know all details of that robot. For example, to move the robot's arms, a ROS command is issued, or a ROS-based Python or C++ script is written by the robot designers cause the robot to respond as commanded. The scripts can, in turn, call various control programs that cause the actual motion of the robot's arms. It is also possible to design and to simulate any complicated robotic application using ROS. In addition, ROS not only applies to the central processing of robotics but also to sensors and other subsystems. ROS hardware abstraction combined with low-level device control are being upgraded toward the latest technology.

ROS is used by many thousands of users worldwide and knowledge can be shared between users. The users range from hobbyists to professional developers of commercial robots to research community. In addition to the large group of ROS researchers, there is a ROS-Industrial group dedicated to applying ROS software to robots for manufacturing. [52]

3.1.1 ROS nodes

Basically, nodes are processes that perform some action. The nodes themselves are really software modules but with the ability to register with the ROS Master node and communicate with other nodes in the system. The ROS design idea is that each node is an independent module that interacts with other nodes using ROS communication protocols. The nodes can be created in various ways, they can be created directly from a Linux terminal window by typing some specific commands. Alternatively, nodes can be created as part of a program written in Python or C programming languages.

3.1.2 Publishing and Subscribing nodes

One of the strengths of ROS is that a particular task, such as controlling a wheeled mobile robot, can be separated into a series of much simpler tasks. These tasks can include the perception of the environment using: a camera or laser scanner, map making, route (or path) planning, monitoring the battery level, etc. Each of these actions might consist of a ROS node or series of nodes to accomplish the specific tasks.

A ROS node code can be independently executed to perform its task; moreover, it can communicate with other nodes by sending or receiving messages. The messages are organized into specific categories named: topics which can consist of data, commands, or other information required for the application being developed. By convention, when ROS sends messages to a topic means the node is publishing to the topic; alternatively, when the ROS node receives messages from a topic, means the node is subscribing to that topic. For instance, the camera provided with a robot is associated with a ROS node publishing repeatably the image on a particular topic. This topic can be used by another node that shows the image on the computer screen. The node that receives the information is said to subscribe to the topic being published. In some cases, a node can both publish and subscribe to one or more topics. In the presented works, many nodes have been used, created, or customized in order simulate the USV consumption.

3.1.3 ROS messages

ROS messages are defined by the type of message and the data format. The ROS package named std_msgs, for example, has messages of type String which consists of a string of characters. Other message packages for ROS have messages used for robot navigation or robotic sensors. In our work, the types of messages that have been used are the odometry messages containing the position vector of the USV as well as its velocity vector.

3.1.4 ROS Master

The ROS nodes are typically small and independent programs that can run concurrently on several systems. Communication is established between the nodes by the ROS Master. The ROS Master provides naming and registration services to the nodes in the ROS system. It tracks publishers and subscribers to the topics. The role of the Master is to enable individual ROS nodes to locate one another. The most often used protocol for connection is the standard Transmission Control Protocol/Internet Protocol (TCP/IP) or Internet Protocol called TCPROS in ROS. Once these nodes are able to locate one another, they can communicate with each other peer-to-peer.

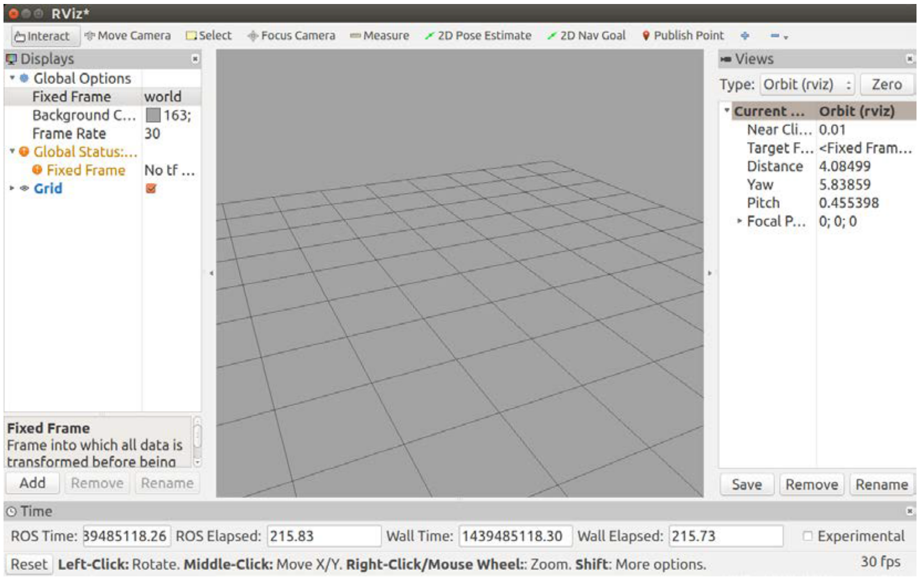
3.1.5 Introducing rqt tools

The rqt tools (ROS Qt GUI toolkit) that are part of ROS allow graphical representations of ROS nodes, topics, messages, and other information. One of the common uses of rqt is to view the nodes and topics that are active. We given later in this thesis, the RQT graph representing different ROS processes being developed and used.

3.2 Rviz software

Rviz, abbreviation for ROS visualization, is a powerful 3D visualization tool for ROS (see Figure 3.1). It allows the user to view the simulated robot model, log sensor information from the robot's sensors, and replay the logged sensor information. By visualizing what the robot is seeing, processing, and doing, the user can debug a robot application from sensor inputs to planned (or unplanned) actions.

Rviz displays 3D sensor data from stereo cameras, lasers, Kinects, and other 3D devices in the form of point clouds or depth images. 2D sensor data from webcams, RGB cameras, and 2D laser rangefinders can be viewed in rviz as image data. If an actual robot is communicating with a workstation that is running rviz, rviz will display the robot's current configuration on the virtual robot model. ROS topics will be displayed as live representations based on the sensor data published by any cameras, infrared sensors, and laser scanners that are part of the robot's system. This can be useful to develop and debug robot systems and controllers. Rviz provides a configurable Graphical User Interface (GUI) to allow the user to display only information that is pertinent to the present task.



Global Options

Fixed Frame	world
Background C...	163;
Frame Rate	30
Global Status:...	
Fixed Frame	No tf ...
Grid	☑

Fixed Frame
Frame into which all data is transformed before being

Type: Orbit (rviz)

Current ...	Orbit (rviz)
Near Cli...	0.01
Target F...	<Fixed Fram...
Distance	4.08499
Yaw	5.83859
Pitch	0.455398
Focal P...	0; 0; 0

ROS Time: 39485118.26 ROS Elapsed: 215.83 Wall Time: 1439485118.30 Wall Elapsed: 215.73 ☐ Experimental

Left-Click: Rotate. **Middle-Click:** Move X/Y. **Right-Click/Mouse Wheel:**: Zoom. **Shift**: More options. 30 fps

Figure 3.1: Rviz window

The central window on the Rviz main screen provides the world view of a 3D environment. Typically, only the grid is displayed in the center window or the window is blank.

The main screen is divided into four main display areas: the central window, the Displays panel to the left, the Views panel to the right, and the Time panel at the bottom. Across the top of these display areas are the toolbar and the main screen menu bar. Each of these areas of the rviz main screen is described in the following sections. This overview is provided so that you can gain familiarity with the rviz GUI.

Rviz-based robot models can be created using Unified Robot Description Language URDF. The latter is an XML format specifically defined to represent robot models down to their component level. These URDF files can become long and cumbersome on complex robot systems. Xacro (XML Macros) is an XML macro language created to make these robot description files easier to read and maintain. Xacro helps you reduce the duplication of information within the file.

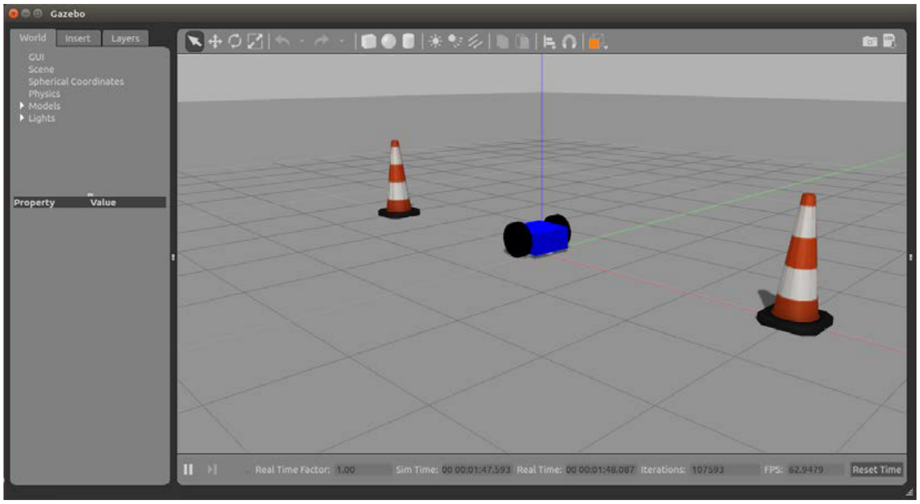
3.3 Gazebo software

Gazebo is a free and open source robot simulation environment developed by Willow Garage. As a multi-functional tool for ROS robot developers, Gazebo supports the following:

- Designing of robot models
- Rapid prototyping and testing of algorithms
- Regression testing using realistic scenarios
- Simulation of indoor and outdoor environments
- Simulation of sensor data for laser range finders, 2D/3D cameras, kinect-style sensors, contact sensors, force-torque, and more

- Advanced 3D objects and environments utilizing Object-Oriented Graphics Rendering Engine (OGRE).
- Several high-performance physics engines (Open Dynamics Engine (ODE), Bullet, Simbody, and Dynamic Animation and Robotics Toolkit (DART)) to model the realworld dynamics.

The Gazebo GUI (see Figure 3.2) is similar to rviz in many ways. The central window provides the view for Gazebo's 3D world environment. The grid is typically configured to be the ground plane of the environment on which all the models are held due to gravity in the environment.



Real Time Factor: 1.00 Sim Time: 00 00:01:47.593 Real Time: 00 00:01:48.087 Iterations: 107593 FPS: 62.9479

Figure 3.2: Gazebo window

3.4 Other ROS simulation environments

The Gazebo software is only one simulator that can interface to ROS and ROS models. A list of other related simulators, both open source and commercial is given as follwos:

- MATLAB with Simulink is a commercially available, multi-domain simulation and modeling design package for dynamic systems. It provides support for ROS through its Robotics System Toolbox [53].

- Stage is an open source 2D simulator for mobile robots and sensors
- Virtual Robot Experimentation Platform (V-REP) is a commercially available robot simulator with an integrated development environment. Developed by Coppelia Robotics, V-REP lends itself to many robotic applications.

3.5 USV simulation environment

As previously reported, USV are now being used in applications such as: search and rescue, containment of oil spills, structural inspection of bridges, tsunami/earthquake forecast, homeland security, environmental monitoring, etc.[1–3]. USVs offer significant advantages over other robotic platforms (aerial and underwater), including payload and energy capacities, localization resources, as well as access to conventional data communication capabilities. In spite of their importance in numerous tasks, this section shows that the ability to simulate USVs remains rare in popular robotics simulation frameworks. One of the main challenges of such simulators is the requirement to tackle complex environmental disturbances such as waves, winds, and water currents. The lack of a standard simulation environment has adverse effects on USV research. Among them, we highlight the absence of a standard modular testing platform for USV Guidance, Navigation and Control (GNC), which makes it challenging to perform comparisons and benchmark these GNC methods. The difference in the way natural disturbances are modeled has important effects on GNC strategies especially in small USVs due to their low inertia and small size. A GNC strategy which does not address environmental disturbances is likely to perform poorly in the field without ideal weather conditions, especially in unpredictable disaster scenarios.

Towards solving this problem, and expecting to assist USV control designers and USV researchers to benchmark their approaches, this section presents an open-source 6-Degree-Of-Freedom simulator for USVs, integrated with the Robot Operating System (ROS)-based framework [1], which models environmental disturbances like winds, water currents, and waves. The proposed tool can simulate different boats, such as those with differential propellers, a single propeller and rudder, air-boats, hovercrafts, and sailboats.

The main features provided the used USV simulators are listed as follows:

- A freely available modular USV simulator where it is possible to model different boats, evaluation scenarios, and GNC strategies. The source code of the simulator or USVsim, is available with an open-source license.
- Improvement of buoyancy effects, influencing USVs' roll, pitch, and yaw.
- Integration of a wind model which affects parts of the boat above the line of water.
- Integration of a hydrological model to simulate water currents of water bodies, applying forces to the boat.
- Ready-to-use USV models and simulation scenarios.

3.5.1 Related software

As given in the literature review chapter, there are many surveys on the topic of computer-based simulators for unmanned systems. About ten open source simulators (Stage, UWSim, Gazebo, "FreeFloating" plugin, V-REP, RobotX Simulator) were evaluated in [1] related to their simulation capabilities. As a result, the classification of the given simulators is illustrated in Table 3.1 according to the following features:

- Waves: indicates the specific ability to simulate 3D waves and offers ways for integration with vehicle simulation (✓✓), i.e., there are waves simulations, but they are for visualization purposes only (✓),or inability (x), to visually simulate waves;
- Buoyancy: Archimedes' principle which makes the USV float over waves. This criterion has been classified into three levels: no buoyancy (x), which means the robot moves over a rigid plane and waves have no influence on the boat; simplified buoyancy (✓), which means it only performs rigid vertical movement of

the entire vessel, according to the height of the water plane; improved buoyancy (✓✓), where the boat tends to follow the shape of the water, resulting in pitch and roll rotations;

- Water currents: describes the method used to apply the water current forces to the vessels: not applied (x); a constant force over the time and space (✓); or a variable force over the time and space (✓✓),according to some Computational Fluid Dynamics (CFD) model;
- Wind currents: similar to the previous item, but the wind force is applied only for objects above the line of water;
- Underwater thruster: how the thruster force is simulated under the water: not simulated (x); a linear function computes the percentage of a maximum force to be applied by the thruster (✓); simulates the dynamics effects of the helix blades of the propeller (✓✓);
- Above water thruster: the same as the previous item, but this thruster is used above the surface of the water, such as in air-boats ;
- Foil: how the foil dynamics are simulated: not simulated (x); simulates nonlinear Lift and Drag forces on the foil (✓✓).

Table 3.1: Physical fidelity of multiple USV simulators

simulator	waves	buoyancy	water current	wind current	thruster underwater	thruster above water	foil
UWSim	✓	✓	x	x	✓	x	x
Gazebo	x	x	x	x	✓	✓	x
Freefloating Gazebo	✓	✓	✓	x	✓	✓	x
V-REP	✓	✓	x	x	✓	✓	x
RobotX Simulator	✓	✓	x	✓	✓	✓	x
USV Sim	✓	✓	✓	✓	✓	✓	✓

The UWSim software is a modular and extensible tool on underwater simulations and Unmanned Underwater Vehicles (UUVs). It offers visually realistic wave and underwater simulations, working mainly as a visualization tool for external modules such as Gazebo, which is responsible for the control algorithms, vehicles, and environment dynamics. It provides ready-to-use scenarios and one UUV named Girona 500 UUV, with the ARM5 manipulator. Its main limitations for boat simulation include: lack of dynamics of a rudder; simplified water current models (i.e., a single constant force is applied to the entire scenario); the wind simulation appears to only affect the wave shape and height, not the vehicle movement; and finally it is not straightforward to model realistic boat behavior, since its focus is clearly on UUVs.

Gazebo software is a popular simulation tool for unmanned systems. It can simulate different robots and complex 3D environments together with the support of several physical simulation engines. Its modular structure enables the extension of its core features through plugins. Most of the current development efforts using Gazebo focus on Unmanned Ground Vehicles (UGVs) and Unmanned Underwater Simulator UUVs. Few works introduce new features designed for USVs, such as the RobotX Simulator and the FreeFloating plugin.

The development of RobotX Simulator started in the early 2018, and its primary objective is to host the Virtual Maritime RobotX Competition (VMRC). The VMRC aims to be the entry point for teams that aspire to participate in the Maritime RobotX Challenge. The simulator builds upon the ROS/Gazebo environment and is under development. It currently provides a model for the WAM-V, a catamaran with differential actuation used for the RobotX competition. This virtual model presents the effects of buoyancy, waves, and also some hydrodynamic effects such as added mass and damping. It also has a simplified wind model (a constant force). The simulator does not provide modules for foil dynamics or realistic wind and water currents. Although the thrusters can be simulated by a linear function of maximum force (defined by the user), this feature can be upgraded by using the LiftDrag plugin of Gazebo Simulator.

FreeFloating is a simulation tool directed for underwater robots used in conjunction with UWSim and Gazebo. It offers a simple buoyancy scheme (applied to the submarine, until it reaches the surface plane Z = 0) and the viscous force of water over UUVs while at the same time enabling the control over wrench, joint states, and body velocity of robots. It can also be combined with the LiftDrag Gazebo plugin to simulate the aerodynamics of thruster blades. However, even when combined with all those modules (see Table 3.1), it does not fully implement rudder models, airboats, or even realistic wind or water currents. .

V-REP software supports add-ons, plugins, socket communication, ROS integration, and Lua scripting. It can use particle simulation to emulate air or water jets, propellers and jet engines, even though, considering the demos and the documentation available on the website, there is no working boat usage example. There is a differential USV model implemented into V-REP with a simple visual effect of waves (no effect on the boat‘s pose) and the vehicle floats over a flat surface. As far as we know, there is no simulation of the influence of water and wind currents on the boat, or a readily available model to describe the rudder dynamics on V-REP.

The study of reviewing existing simulation environments for USV revealed the best features and limitations of each of them. The used software is an improved simulator reusing the following tools:

- Gazebo: as the core simulation engine, due to its modular design based on plugins, dynamics collision simulation, number of modeled sensors, USVs can be designed using XML-based URDFXacro format, and the ease of development given the large community of users and code maintainers;
- Free-floating Plugin: Despite some limitations detailed later, the structure of this plugin, originally developed for underwater robots, was re-purposed to be used for USV simulation and we improved the hydrodynamics and buoyancy effects compared to the original plugin;

- UWSim: for water and wave visual effects and due to sensors that are readily available for use;
- LiftDrag Gazebo Plugin: we mainly reuse its structure to calculate foil dynamics in the parts of the boats.

3.5.2 System architecture

This section presents the used simulator's architecture and the main contributions designed on top of the available simulation resources presented in this chapter. Figure 3.3 gives an overall description of the system architecture where the blue boxes represent the added simulation modules.

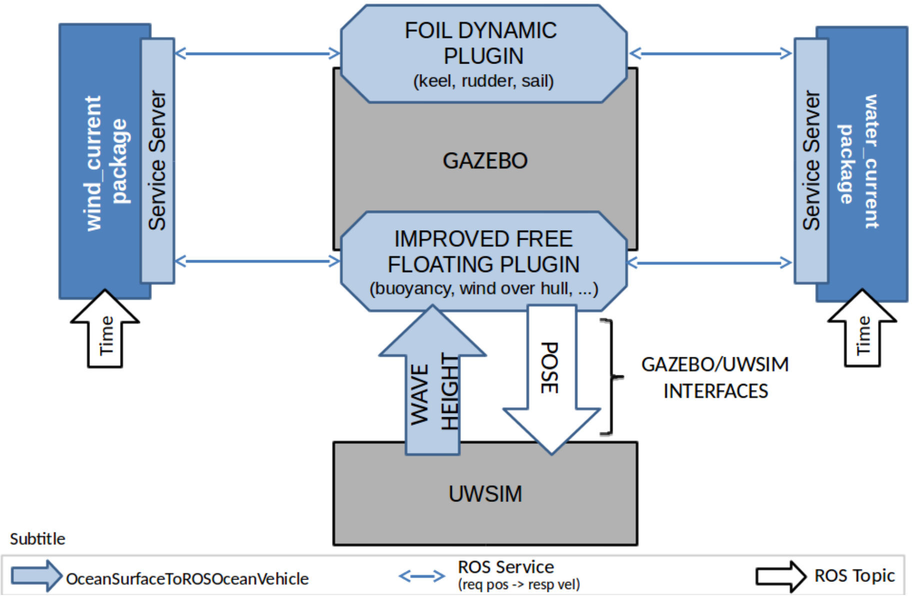


Figure 3.3: USV simulator architecture

Gazebo is used as the main simulation engine while UWSim is used for visualization purposes. As represented in Figure 3.3, the core of Gazebo is not modified, but we included one new plugin called usv_foil_dynamics and also the improved FreeFloating. UWSim has been modified to provide a service where the plugins request the wave height at a specific position of the map. The water and wind current generators are modeled as ROS nodes which receive requests from improved

FreeFloating Gazebo to enhance the boat motion realism through wind and current information. Besides, the windwater current generator is used by the foil dynamics plugin to compute the forces which are directly applied to the foil (if the boat has one), using as input the velocities of the foil and the velocity of the windwater current. All modules have their corresponding YAML files where the user can set specific parameters without changing the source code.

According to [21] the main forces that govern a USV movement provided within the simulator are:

- Hydrodynamic forces (τ_{hyd}): added mass (the virtual mass added to the boat by the mass of water moved with the boat), potential damping and viscous damping;
- Hydro-static forces (τ_{ths}): restoring forces (buoyancy);
- Wind forces (τ_{wind});
- Wave forces (τ_{waves}) ;
- Control and propulsion forces (τ) involving: foil dynamics (rudder, keel and sail) and thrusters

3.5.3 Proposed robot models in USV simulator

Four different ready-to-use USV models, as illustrated in Figure 3.4, are integrated into the given simulator as default in order to describe different dynamics of USV: an airboat, a motorized boat with one rudder, a differential boat with two thrusters, and a sailboat. The airboat uses a fan above the surface of the water as a propeller. This fan also rotates on its own axis to change the airboat direction of movement. The airboat suffers more from the effect of drift (sideslip angle) since it does not have any foil underwater. The motorized boat uses an underwater thruster for propulsion and a rudder for changing the movement direction. The differential boat uses two underwater thrusters, which enables the rotation over its axis. The sailboat uses a rigid sail for propulsion, a rudder for changing the movement direction, and it also has a keel to help with sway and roll stability.

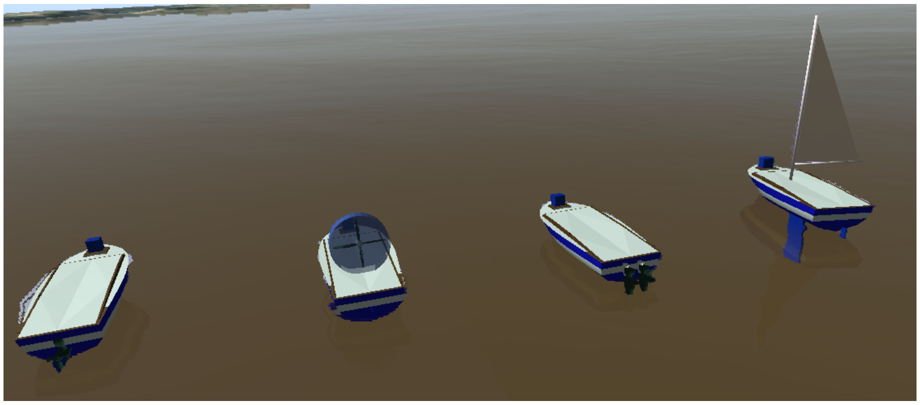


Figure 3.4: Four USV types provided with the simulator

The presented boats can have a variety of sensors thanks to Gazebo, which also offers large documentation on how to model personalized sensors such as water current, temperature, pH, etc. Currently, the ready-to-use USV models have position sensors, Inertial Measurement Unit (IMU) and a laser range finder.

3.5.4 Physical boats Specification

The airboat and differential boats were modeled according to specifications of the Lutra Airboat and Lutra Prop boats respectively, acquired from Platypus. Lutra Airboat and Lutra Prop boats have the same hull dimensions and shape, both are 106 centimeters long, 48 centimeters wide and 15 centimeters tall. While Lutra Airboat weights 9 Kg, Lutra Prop weights 9.7 Kg. In the work given in [1], both boats carried approximately 3 Kg of extra payload, which was needed to properly transport a Real Time Kinematic (RTK) system (RTK antenna, RTK module, power bank, antenna pole, cellphone with internet access). Lutra Airboat is driven by a propeller coupled to the hull's stern, providing forces parallel to the boat's hull. On the other hand, the Lutra Prop boat has two underwater propellers, both are attached such that there is an angle of 15 degrees between the horizontal line and the propeller's axis. The main physical specifications of both boats are presented in

Table 3.2. Most of them are collected by some laboratory measurements. Thruster force, linear drag coefficient and maximum speed are estimated by field trials.

Table 3.2: Lutra Airboat and Lutra Prop parameters.

Parameter	***Lutra Airboat***	***Lutra Prop***
Length	106 cm	106 cm
Width	48 cm	48 cm
Height	45 cm	15 cm
Hull volume	0.02 m	0.02m
Weight	9 Kg	9.7 Kg
Extra payload	3 Kg	3 kg
Thruster Force	3.1 N	22.54 N
Linear drag	6.9	11.33 N
Maximum speed	0.67 m/s	1.41m/s

3.5.5 Water and Wind Current Modules

Water current simulation process

An actual scenario is used as study case in [1]. This place is located at 3002050.5” S5113057.7”W, in the city of Porto Alegre, Brazil, nearby the Dilúvio’s river mouth. This scenario is built based on digital terrain models provided by the Municipality of Porto Alegre. In order to allow water currents to change across space and time more realistically we developed a new ROS package called water_current, which loads data exported from the Hec RAS hydrological simulator. Hec RAS can model one-dimensional (1D) steady flow, as well as one and two-dimensional unsteady hydrologically-based flow calculations in rivers and canals. Thus users can simulate the flow of rivers by inserting simple height maps from terrain and river bed, then exporting Hierarchical Data Format (HDF) files, which store the water velocities for each time step of the simulated water flow. Figure 3.5 shows the resulting water current velocity map, where the water flows to the left-hand side of the image.Note that the narrow channels have higher speeds. By using the HDF files as input for water_current plugin, our simulation architecture requests the velocity of the water at each of the boat's position of links. This is done by a ROS service, where the improved FreeFloating Gazebo sends the (x,y) position of each link (see Figure 3.5) and the water_current answers with a flow velocity for each point.

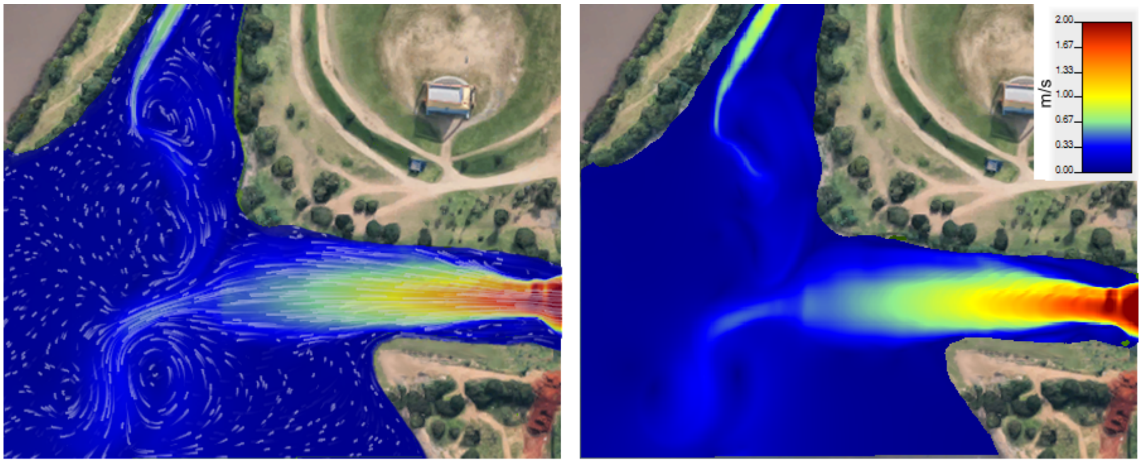


Figure 3.5: Hec RAS simulation of water current for a given environment showing the intersection of two rivers. The image on the left shows white particle trails moving on the water. On the right, the fastest flow is depicted in red, while the slowest is in blue.

Wind speed simulation process

The wind_current ROS package loads wind data exported by the CFD software OpenFoam (Open source Field Operation And Manipulation) [1]. Openfoam is composed of a C++ library capable of solving specific continuum mechanics problems and it provides several utilities to prepare a mesh for simulation, process results, and so on. It generates wind simulation based on a 3D terrain model with multiple obstacles, such as buildings and bridges, and by specifying the simulation parameters, like wind velocity, simulation duration, and time step. The resulting simulation (see Figure 3.6-8) can be analyzed and exported to the wind_current module by an OpenFoam utility named paraView. The data exported by paraView contains the wind velocity at each time step, that can be loaded into the wind_current module and accessed in the simulation architecture by a ROS service. To do that, the improved FreeFloating Gazebo sends the vehicle's position of each link, the wind_current package answers with wind velocity for each point. Figure 3.7 shows the resulting wind model for the given test scenario, including examples of wind models around the bridge and buildings.

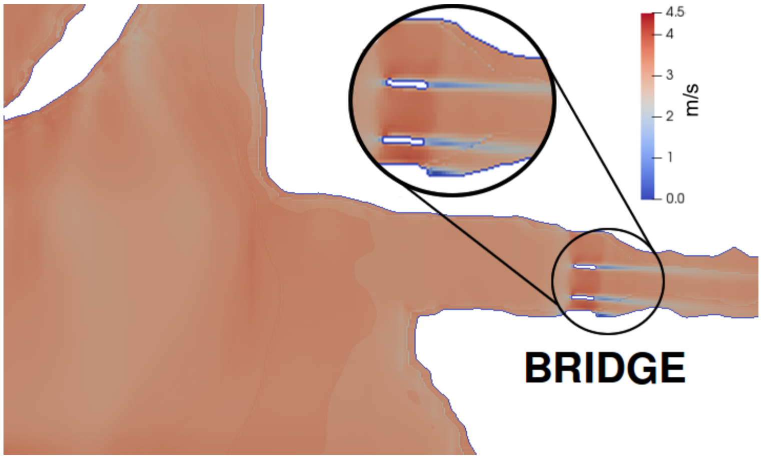


Figure 3.6: View of the same region presented in Figure 3.5, including details of the wind near by the bridge

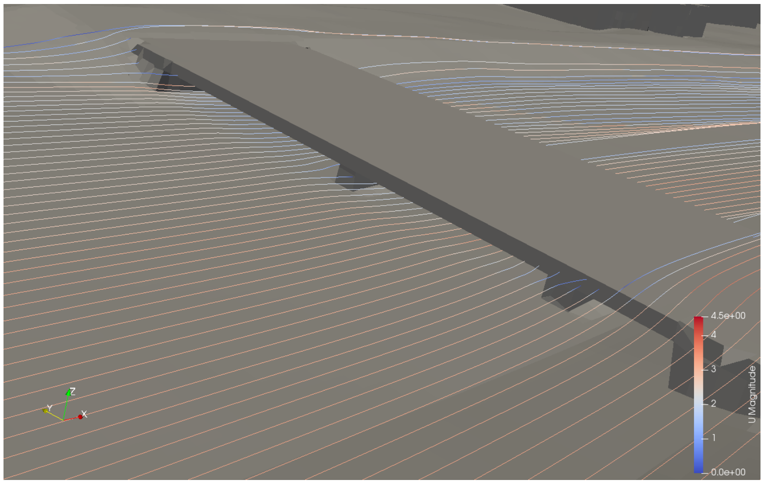


Figure 3.7: View of wind around the bridge

3.5.6 Comparison between Physical and Simulated Boats

To move the boats straight ahead, each boat's drive is configured with the maximum PWM values (2000 for Airboat and 1700 for Differential). This enables us to obtain the velocity time response of both simulated and real boat, as presented next [1]. The physical boats were equipped with Emlid's Reach RTK Global Positioning System (GPS). The GPS was configured to received the corrections from a public

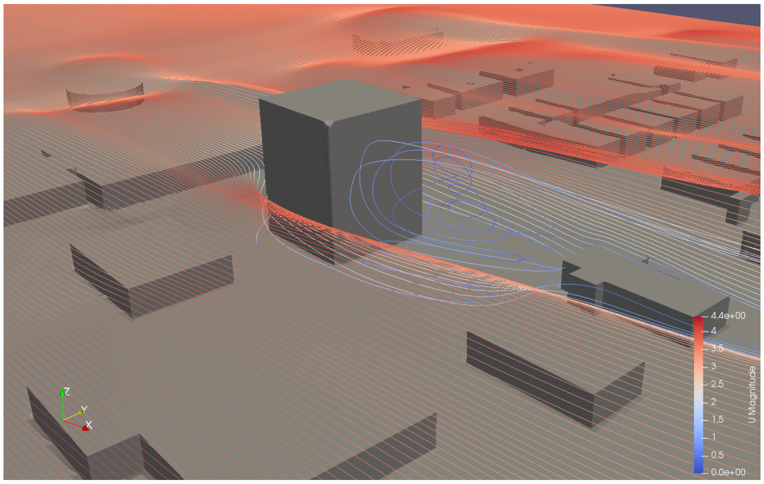


Figure 3.8: View of wind around nearby buildings

base station provided by IBGE, located at 3004026.700 S 5107011.200 W, via the NTRIP protocol (Networked Transport of Radio technical commission for marine service—RTCM—via Internet Protocol). The lake used for the test is located at 3004057.900 S5103023.900 W, and it has an area of about 85 m by 55 m. The distance from the base station to the lake is about 6.1 km, guaranteeing optimal GPS corrections. During the tests, the GPS got more than 99% of fix (best GPS signal), generating excellent accuracy. Finally, there was no significant wind at the test location during the tests. All these characteristics of the test location and test moment were selected to increase the localization accuracy of the physical boats. The simulated boats, on the other hand, have the advantage of a perfect localization, easing the velocity analysis presented next. Figure 3.9 and Figure 3.10 show the comparison of the speed profile obtained in the field trial and simulation for both the Airboat and the Differential boat. Both charts show speed (m/s) per time (in seconds), assuming that time zero was the moment the boats started to move. However, the physical boats have a starting velocity close to zero due to no significant wind and water current that day. During the tests, although looking at the boat we thought that it was still, the GPS RTK captured a small velocity of about 0.1 m/s. For this reason, the starting velocity of the real boats (dotted blue line) was not zero. These charts show that both simulated and real boats were accurate in terms of velocity variation over the time and also accurate in terms of maximal velocity. The mean absolute error and standard deviation for the airboat are 0.0312 m/s and 0.0481 m/s (respectively), so 94.7% of errors are in the range of one standard deviation. While the mean absolute error and standard deviation for the differential boat are 0.0507 m/s and 0.0742 m/s (respectively), so 96.6% of errors are in the range of one standard deviation.

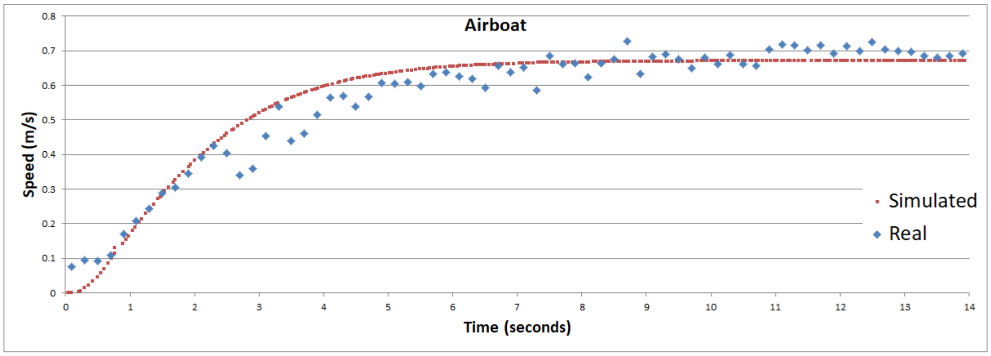


Figure 3.9: Lutra Airboat vs simulated Airboat. PWM of 2000 is used for this chart.

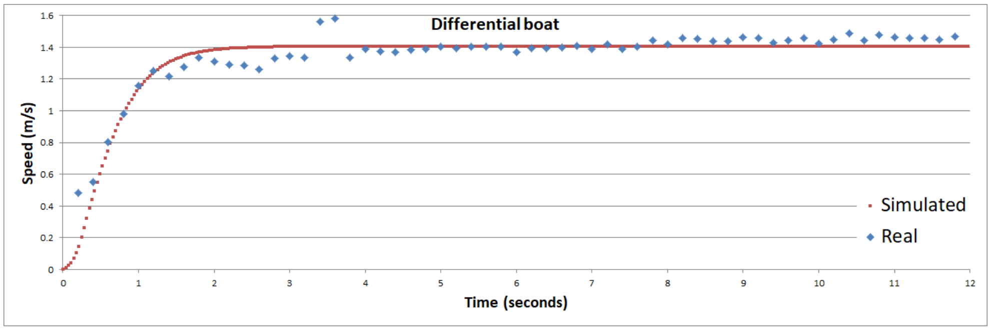


Figure 3.10: Lutra prop vs simulated Differential boat. PWM of 1700 is used for this chart

3.5.7 Differential drive boat at variable disturbances

This section describe an experiment showing the differential boat crossing the river with the proposed disturbances [1], as illustrated in Figure 3.11. The figure shows the unique ability of the proposed simulator to disturb the boat's position and speed according to a hydrological model of the water body, In addition, the boat starts at the righthand side and goes to the top left side of the map. The boat has a higher speed when it is leaving the straight canal, where the water current is faster, and the speed decreases as the boat reaches the open waters on the left-hand side. The total traveled distance is 452.38 m. The boat took 376.98 s (average speed of 1.2 m/s with a peak speed of about 3.04 m/s) to complete the trajectory since it is running in the same direction of the water and wind flows, helping the boat to gain speed.

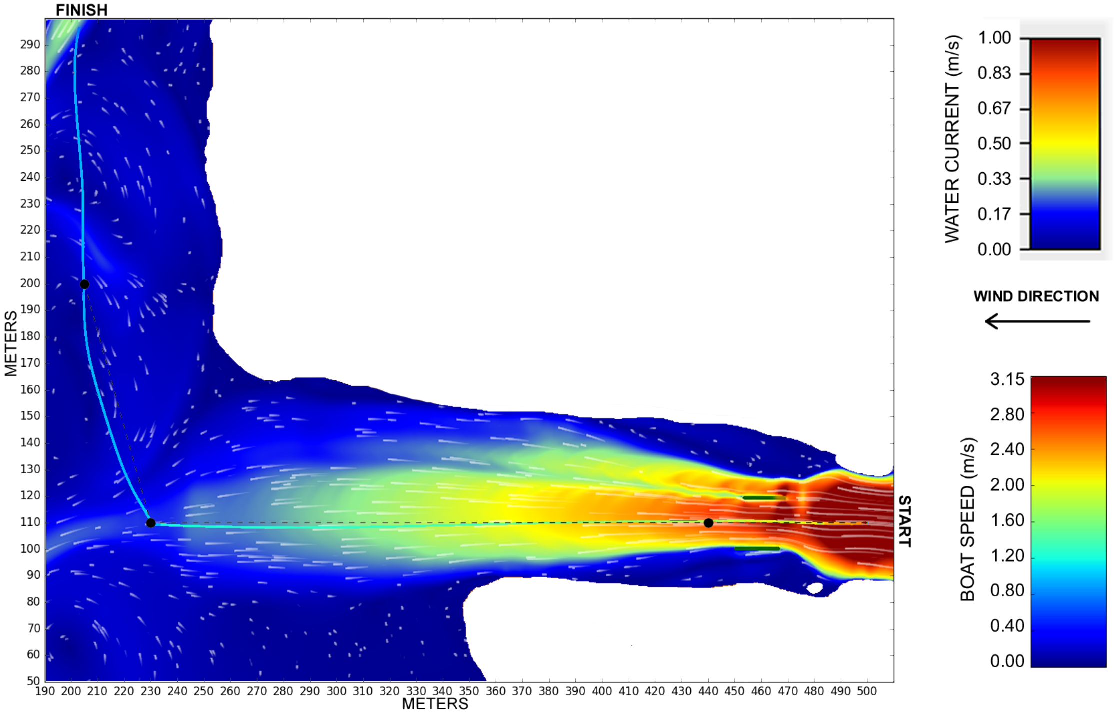


Figure 3.11: Trajectory of a differential boat in the environment. The trajectory color represents the boat speed along the path.

3.5.8 Different Water Speeds of the River

This section shows that the simulator is very coherent with some basic river navigation guidelines. For instance, boats that sail upstream should stay close to the river bank, where the water current is slower than in the middle of the river. On the other hand, boats that sail downstream should stay in the middle, as they can advantage of the river flow, saving fuel. Figure 3.12 represents two different trajectories taken by the differential boat. The pink line, close to the river banks, represents a trajectory where the water current is slower. The red line, in the middle of the river, represents a trajectory where the water current is faster. Both trajectories are starting at the left hand side, thus the water current is in the opposing the direction of the boat.

The pink trajectory took 144.9 s to reach the destination, with an average speed of about 1.10 m/s. The red trajectory took 187.2 s to completion, resulting in an average speed of about 0.85 m/s.

The second part of Figure 3.12 also shows the boat speed for both trajectories during the time. It can be seen that the river bank trajectory, represented by the pink line, keeps a fairly constant speed of about 1.1 and 1.3 m/s. However, the middle river trajectory represented by the red line, has a clear speed reduction as the boat gets closer to the area with faster water currents. The speed starts with about 1.2 m/s but it ends with a speed below 0.8 m/s. Both trajectories have sudden reductions in speed. This is an effect of the controller, when it steers the boats. The boat close to river banks presented fewer speed reductions than the boat in the middle, since the water close to riversides presents less turbulence and small speed than on the middle river water.

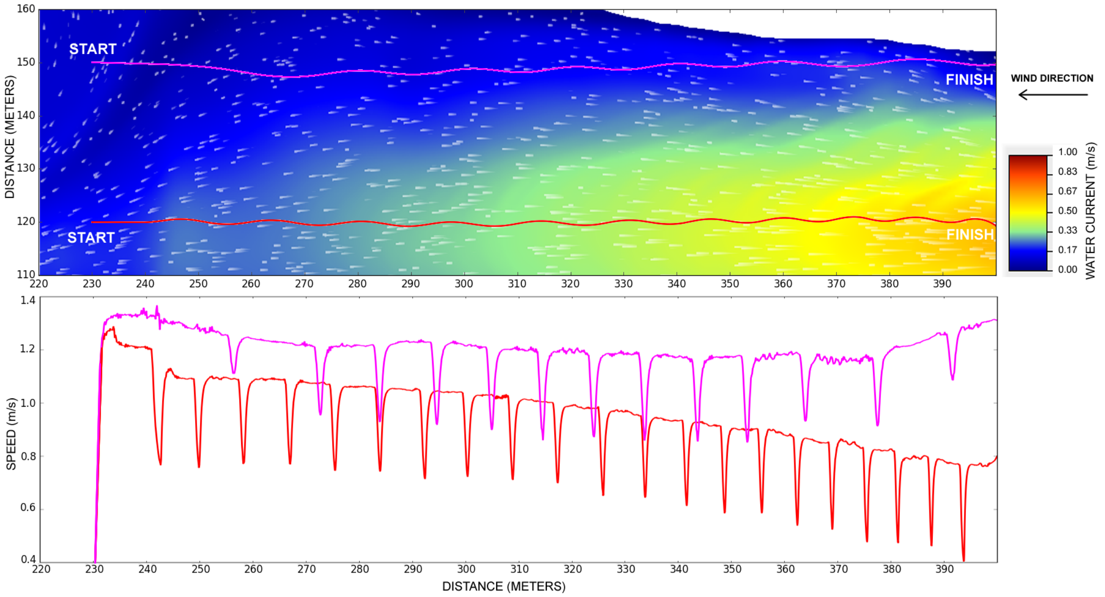


Figure 3.12: Different navigation speeds of the differential boat going upstream, against the river current

Consequently, the presented simulation tools make a considerable contribution to the development of unmanned vehicles permitting to solve many issues in their research fields; however, and as mentioned before, the energy consumption parameter has not been introduced within the simulation engines despite its considerable importance. The presented simulation environment is enriched, as indicated in the following chapters, with energy model and management tools allowing the virtual estimation of the power instantaneously as a function of the USV's speed and environment. Therefore, the total energy required to finish a given scenario can be estimated numerically without any hardware requirement.

Chapter 4

USV dynamics and consumption modelling

4.1 Introduction

The presented chapter illustrates the dynamic behaviour of USV based on marine crafts and hydrodynamics motion and control equations given in [21], as well as the analytical energy and power consumption development which represents a considerable contributions reported in this thesis.

Basically, a marine drone can move in real scenarios either longitudinally, laterally, or vertically, it can also move circularly around each axis producing roll, pitch, and yaw movements. This type of motion system is known as six-degrees-of-freedom (6-DOF) system. In order to simplify our analysis, the marine's wave effect is neglected; thus, the vertical motion, roll rotation, and pitch rotation are ignored because the energy consumed in these three dimensions is much lower compared to the other ones. Therefore, the system is reduced to a 3-DOF motion system by considering only surge velocity u (longitudinal motion), sway velocity v (sideways motion), and yaw rate r (rotation around the vertical axis).

The motion velocity vector (u,v,r) can be represented in two reference frames as illustrated in Figure 5.1:

- Earth-fixed reference frame $\{e\}$ which is denoted by o_e ; x_e; y_e; z_e fixed to the Earth. The positive y_e axis points towards the East, the position x_e towards the North and positive z_e towards the centre of the Earth. The origin denoted by o_e is located on the mean surface of the water at a specified placement.
- Body-fixed reference frame $\{b\}$, The b-frame is fixed to the hull, this refers to o_b; x_b; y_b; z_b. The location of o_b is usually chosen to intercept with the axes of inertia.

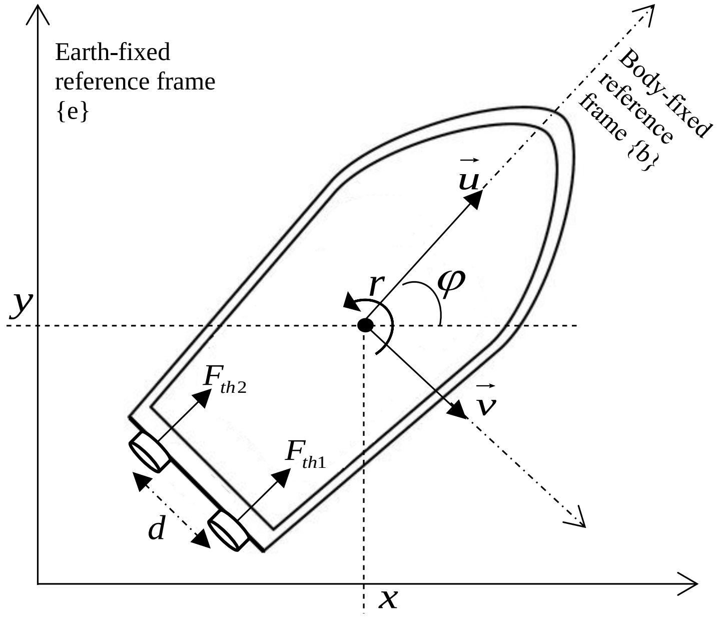


Figure 4.1: Differential Drive USV dynamics representation in $\{e\}$ and $\{b\}$ frames.

In this chapter, we define first the 3-DOF USV dynamic model by considering the environment disturbances (mainly: wind and water current) that is used to develop the power consumption model. In the ships' dynamics analysis, the 3-DOF equations are usually adopted to describe their horizontal motion which is the case when ignoring marine's wave effect. We present in the first section of this chapter the analytical model of the the USV dynamics without disturbances; thereafter, we introduce the mathematical representation of the wind force and the water current in given previously developed model.

4.2 3-DOF USV dynamics without disturbances

In order to exploit the physical properties of the sea-keeping and manoeuvring models, the equations of motions representing the relationship between the the boat's speed vector and a set of forces applied on its shape are represented in a vector form to reduce the number of coefficients in order to simplify calculations and the analysis [21]. Thus, the complete 3-DOF dynamic model of the USVs considered in our thesis is given given Eq.(5.1) represented in $\{b\}$ frame.

$$M.\dot{V} + C(V).V + D(V).V = \tau_{thrust} + \tau_{disturbance} \tag{4.1}$$

where,

$$M = \begin{bmatrix} m - X_{\dot{u}} & 0 & -m.y_g \\ 0 & m - Y_{\dot{v}} & mx_g - Y_{\dot{r}} \\ -my_g & mx_g - N_{\dot{v}} & I_z - N_{\dot{r}} \end{bmatrix},$$

$$C(V) = \begin{bmatrix} 0 & 0 & -m(x_g r + v) + Y_{\dot{v}} v + \frac{Y_{\dot{r}} + N_{\dot{v}}}{2} r \\ 0 & 0 & (m - X_{\dot{u}})u \\ m(x_g r + v) - Y_{\dot{v}} v - \frac{Y_{\dot{r}} + N_{\dot{v}}}{2} r & -(m - X_{\dot{u}})u & 0 \end{bmatrix},$$

and

$$D(V) = D + D_n(V) = - \begin{bmatrix} X_u & 0 & 0 \\ 0 & Y_v & Y_r \\ 0 & N_v & N_r \end{bmatrix} - \begin{bmatrix} X_{u|u|}|u| & 0 & 0 \\ 0 & Y_{v|v|}|v| + Y_{v|r|}|r| & Y_{r|v|}|v| + Y_{r|r|}|r| \\ 0 & N_{v|v|}|v| + N_{v|r|}|r| & N_{r|v|}|v| + N_{r|r|}|r| \end{bmatrix}.$$

The vector $V = [u,v,r]^T$ represents the velocity expression with respect to $\{b\}$ frame as given in Figure 5.1. The components (u,v,r) are the coordinates of the velocity vector V representing the surge velocity, sway velocity, and yaw angular velocity respectively.

M represents the 3-DOF inertia matrix including the added mass parameters where m is the total mass. The vector components (x_g,y_g) represent the coordinates of the gravity centre point of the USV in $\{b\}$ frame which approximately equal to zero and can be ignored.

C is the 3-DOF Coriolis and centripetal matrix. This latter is due to the rotation of the body-fixed reference frame $\{b\}$ with respect to the inertial reference frame $\{e\}$. In addition, D represents the 3-DOF hydrodynamic drag or damping matrix. The different damping terms contribute to both linear and quadratic damping. However, it is in general difficult to separate these effects. In many cases, it is convenient to write total hydrodynamic damping as the summation of the linear drag term D and the non-linear drag term $D_n(v)$. [21]

The term τ_{thrust} represents the thrust forces caused by the propellers represented in the $\{b\}$ reference frame; furthermore, the mathematical models of wind and waves forces and moments are considered to improve the performance and robustness of the analytical model in extreme conditions which is represented in the equation by the term $\tau_{disturbance}$.

Moreover, The terms $X_{\dot{u}}$, $Y_{\dot{v}}$, $Y_{\dot{r}}$, $N_{\dot{v}}$, $N_{\dot{r}}$, $I_{\dot{z}}$, represent the hydrodynamic added mass, whereas the terms X_u, Y_v, Y_r, N_v, N_r represent the linear damping coefficients. both sets can be considered as the model's corresponding hydrodynamic coefficients that can be identified using several approaches as given in upcoming chapters.

In our thesis, it is supposed that the USV is operating as a differential drive boat controlled by two propellers (thrusters) placed on the rear portion of the boat (see Figure 5.1). The two thrusters are used to deferentially control the boat's speed and direction. Thus, the resultant thruster force vector for such a mechanical systems can be expressed as given in Eq.(5.2): [42, 27]

$$\tau_{thrust} = \begin{bmatrix} \tau_u \\ 0 \\ \tau_r \end{bmatrix} = \begin{bmatrix} F_{Th1} + F_{Th2} \\ 0 \\ (F_{Th1} - F_{Th2}).\frac{d}{2} \end{bmatrix} \tag{4.2}$$

As a result, the value τ_u represents the longitudinal thrust while the term τ_r represents the yaw moment. The terms F_{Th1} and F_{Th2} represent the left and the right thrusters' forces (thrust) respectively along $x - axis$ given in $\{b\}$ frame. In addition, the term d represent the distance between the center-lines of each thruster (see Figure 5.1).

In order to mathematically simplify the proposed USV dynamic model, we consider in our thesis the following hypotheses: [40, 42]:

1. The maximum speed of the USV does not exceed $1.5m/s$; thus, the effect of the non-linear drag term of the hydrodynamic drag matrix can be neglected i.e. $D_n(v) = O$.
2. The off-diagonal terms are much smaller than the diagonal terms of the matrices M and D; hence, the off-diagonal terms of both matrices can be ignored.
3. The coincident center of the added mass and gravity N_v can be replaced by Y_r. A combination of approximate fore-aft symmetry and light draft suggests that the sway force arising from yaw rotation and the yaw moment induced by the acceleration in the sway direction are much smaller than the inertial and added mass terms. Therefore, we assume in the Coriolis and centripetal matrix $C(v)$ that $N_v = Y_r = 0$.

As a result, the simplified 3-DOF dynamic model of the USVs is deduced and represented as given in Eq.(5.3).

$$M.\dot{V} + C(V).V + D(V).V = \tau_{thrust} + \tau_{disturbance} \tag{4.3}$$

where,

$$M = \begin{bmatrix} m - X_{\dot{u}} & 0 & 0 \\ 0 & m - Y_{\dot{v}} & 0 \\ 0 & 0 & I_z - N_{\dot{r}} \end{bmatrix} = \begin{bmatrix} m_{11} & 0 & 0 \\ 0 & m_{22} & 0 \\ 0 & 0 & m_{33} \end{bmatrix},$$

$$C(v) = \begin{bmatrix} 0 & 0 & -(m - Y_{\dot{v}})v \\ 0 & 0 & (m - X_{\dot{u}})u \\ (m - Y_{\dot{v}})v & -(m - X_{\dot{u}})u & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & -m_{22}v \\ 0 & 0 & m_{11}u \\ m_{22}v & -m_{11}u & 0 \end{bmatrix},$$

and

$$D(v) = -\begin{bmatrix} X_u & 0 & 0 \\ 0 & Y_v & 0 \\ 0 & 0 & N_r \end{bmatrix} = \begin{bmatrix} d_{11} & 0 & 0 \\ 0 & d_{22} & 0 \\ 0 & 0 & d_{33} \end{bmatrix}.$$

Besides, since the given model is expressed in $\{b\}$ reference frame, some appropriate kinematic transformations between the two references $\{e\}$ and $\{b\}$ should be derived. Let's consider that η and $\dot{\eta}$ are position vector and velocity vector respectively in $\{e\}$ reference frame.

Thus, the velocity vector represented in $\{e\}$ frame is given by Eq.(5.4)[40, 42].

$$\dot{\eta} = [\dot{x},\dot{y},\dot{\varphi}]^T = J(\eta).v \tag{4.4}$$

Where the vector η is the position vector of the USV in $\{e\}$ frame given by Eq.(5.5).

$$\eta = [x,y,\varphi]^T \tag{4.5}$$

The terms x and y are the USV's position coordinates in $x - axis$ and $y - axis$ respectively in $\{e\}$ frame; moreover, φ represents its yaw angle (or angle of attack of the USV). The term $J(\eta)$ represents the transformation matrix [42, 43] given by Eq.(5.6). Thus, the velocity vector in $\{e\}$ frame is deduced as given by Eq.(5.7).

$$J(\eta) = \begin{bmatrix} \cos\varphi & \sin\varphi & 0 \\ \sin\varphi & -\cos\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.6}$$

$$\dot{\eta} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} u\cos\varphi + v\sin\varphi \\ u\sin\varphi - v\cos\varphi \\ r \end{bmatrix} \tag{4.7}$$

To identify the analytically developed USV model's parameters and to verify its effectiveness for the case of study surface drone, the expanded form representations is derived form Eq.(5.3) and Eq.(5.7). This expanded form is given in Eq(5.8-11).

The remaining details of the model parameters identification approaches is given later in chapter 5.

$$\begin{cases} \dot{x} = u\cos\varphi + v\sin\varphi \\ \dot{y} = u\sin\varphi - v\cos\varphi \\ \dot{\varphi} = r \end{cases} \tag{4.8}$$

$$(m - X_{\dot{u}})\dot{u} - (m - Y_{\dot{v}})vr + X_u u = F_{Th} = F_{Th1} + F_{Th2} \tag{4.9}$$

$$(m - Y_{\dot{v}})\dot{v} + (m - X_{\dot{u}})ur + Y_v v = 0 \tag{4.10}$$

$$(I_z - N_{\dot{r}})\dot{r} + (X_{\dot{u}} - Y_{\dot{v}})uv + N_r r = F_N\frac{d}{2} = (F_{Th1} - F_{Th2})\frac{d}{2} \tag{4.11}$$

4.3 USV Dynamics with water current disturbances

Actually, the USVs operate on environments having considerable water current disturbances with different directions and speeds. However, the water current's speed and directions can be either helpful or obstructive to USV motion; and therefore, the water current speed makes a significant impact on the USV behaviour,; hence, on its energy consumption. The analytical modelling of the water current disturbances can be done by differentiating the actual USV velocity vector and the water current velocity vector producing the USV's relative speed which mathematically represented as given by Eq.(5.12) [21].

$$\dot{\eta}_r = \dot{\eta} - V_c \tag{4.12}$$

The term $\dot{\eta}_r$ represents the relative velocity vector of the USV represented in $\{e\}$, V_c is the water current vector represented in $\{e\}$ frame as well that can be represented using the vector:$V_c = [V_{cx},V_{cy},0]^T$.

$$\dot{\eta}_r = \begin{bmatrix} u\cos\varphi + v\sin\varphi - V_{cx} \\ u\sin\varphi - v\cos\varphi - V_{cy} \\ r \end{bmatrix} \tag{4.13}$$

In order to associate the relative velocity of the USV with the previously described model, the relative speed should be represented in the same reference frame (i.e. $\{b\}$); thus, the relative velocity can be represented in the $\{b\}$ reference frame by using by Eq.(5.4) and the inverse of the transformation matrix as given in Eq.(5.14).

$$\begin{aligned} \dot{\eta}_r &= J(\eta).v_r \\ J^{-1}(\eta).\dot{\eta}_r &= J^{-1}(\eta).J(\eta).V_r \\ V_r &= J^{-1}(\eta).\dot{\eta}_r \end{aligned} \tag{4.14}$$

where: V_r is the USV's relative velocity given in $\{b\}$ frame and can be represented using the vector: $V_r = [u_r,v_r,r_r]^T$. In addition, $J^{-1}(\eta)$ represents the inverse transformation matrix given in Eq.(5.15).

$$J^{-1}(\eta) = \begin{bmatrix} \cos\varphi & \sin\varphi & 0 \\ \sin\varphi & -\cos\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} \cos\varphi & \sin\varphi & 0 \\ \sin\varphi & -\cos\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.15}$$

Therefore, the relative velocity vector of the USVs represented in $\{b\}$ frame is deduced as a function of the boat's speed vector and the water current vector as represented in Eq.(5.16).

$$\begin{aligned} V_r = \begin{bmatrix} u_r \\ v_r \\ r \end{bmatrix} &= \begin{bmatrix} \cos\varphi & \sin\varphi & 0 \\ \sin\varphi & -\cos\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} u\cos\varphi + v\sin\varphi - V_{cx} \\ u\sin\varphi - v\cos\varphi - V_{cy} \\ r \end{bmatrix} \\ &= \begin{bmatrix} u - V_{cx}\cos\varphi - V_{cy}\sin\varphi \\ v - V_{cx}\sin\varphi + V_{cy}\cos\varphi \\ r \end{bmatrix} \end{aligned} \tag{4.16}$$

As a consequence, and based on the USV dynamic model expressed in Eq.(5.3), the thrusters' force vector can be expressed actually as function of the USV's relative velocity as given in Eq.(5.17). On the other hand, we illustrate in the upcoming section the approach used to establish the wind disturbances force and to associate it with the USV dynamic model.

$$\tau_{thrust} = M.\dot{V}_r + C(V_r).V_r + D(V_r).V_r - \tau_{wind} \tag{4.17}$$

4.4 USV Dynamics with wind disturbances

The wind disturbances located on the surface of oceans or rivers provide considerable lateral and frontal forces exerted on the USV body. Like water current effect, the wind disturbances can be either helpful or obstructive to the USV motion according to its direction and to that of which makes obviously a considerable influence on the USV consumption. To study the wind effect on the USV energy consumption, both lateral and frontal wind forces applied on the lateral and frontal areas of the non-submerged part of the boat respectively are modeled and integrated into the previously developed dynamic model given in Eq.(5.17).

Let $V_w = [V_{wx},V_{wy}]^T$ be the non-rotational wind velocity vector expressed in $\{e\}$ frame, and $V_w = [V_w,\gamma_w]^T$ represents its vector in $\{b\}$ reference frame as shown in Figure 5.2.

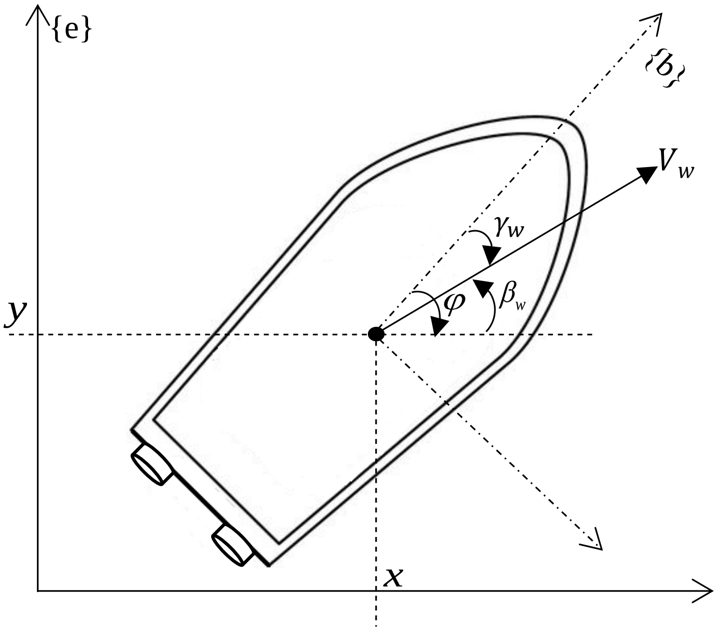


Figure 4.2: Wind speed vector representation in $\{e\}$ and $\{b\}$ frames.

In the three-Degrees-of-freedom (3-DOF) USV motion system, the wind disturbances create a 3-DOF force exerted on the centre of the gravity of the boat which can be represented by the vector: $\tau_{wind} = [X_{wind},Y_{wind},N_{wind}]^T$. The wind force vector is expressed according to [21] as a function of USV's non-submerged frontal and lateral areas exposed to the wind, in addition to some coefficients to be identified in the upcoming chapters. The resulting wind force expression is given by Eq.(5.18).

$$\tau_{wind} = \begin{bmatrix} X_{wind} \\ Y_{wind} \\ 0 \end{bmatrix} = \begin{bmatrix} q.C_x(\gamma_w).A_{Fw} \\ q.C_y(\gamma_w).A_{Lw} \\ 0 \end{bmatrix} \tag{4.18}$$

The term γ_w represents the angle of attack of the wind with respect to the USV in $\{b\}$ frame such that: $\gamma_w = \varphi - \beta_w$. The term q represents the dynamic pressure of the apparent wind given by:

$$q = \frac{1}{2}\rho_a V_w^2$$

[21]

Moreover, the angle β_w represents the wind direction given in $\{e\}$ reference frame. A_{Fw} and A_{Lw} are the frontal and lateral non-submerged areas respectively. ρ_a is the air density which approximately equals to $1.184 Kg/m^3$ when $T = [10°C, 25°C]$.

Basically, For ships that are symmetrical with respect to xz plane, the wind coefficients $C_x(\gamma_w)$ and $C_y(\gamma_w)$ for horizontal plane motions can be approximated according to [21] by the formulas: $C_x(\gamma_w) = c_x \cos(\gamma_w)$, $C_y(\gamma_w) = c_y sin(\gamma_w)$. Where: c_x and c_y are some constant values that depend on the physical shape of the surface drone.

Consequently, the force exerted on USV center of mass create by the wind disturbances represented in $\{b\}$ reference frame is analytically expressed as a function of the wind's speed and direction as given in Eq.(5.9).

$$\begin{aligned} \tau_{wind} = \begin{bmatrix} X_{wind} \\ Y_{wind} \\ N_{wind} \end{bmatrix} &= \begin{bmatrix} \frac{1}{2}\rho_a.A_{Fw}.c_x.V_w^2 \cos(\gamma_w) \\ \frac{1}{2}\rho_a.A_{Lw}.c_y.V_w^2. \sin(\gamma_w) \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} \frac{1}{2}\rho_a.A_{Fw}.c_x.V_w^2. \cos(\varphi - \beta_w) \\ \frac{1}{2}\rho_a.A_{Lw}.c_y.V_w^2. \sin(\varphi - \beta_w) \\ 0 \end{bmatrix} \end{aligned} \tag{4.19}$$

4.5 USV consumption modelling

As previously reported, the power consumption of the USVs can be split into two parts: Electrical part which involves the static power due to static electrical device consumption (on-board computer, power losses, communication circuits, sensors, etc), this type of power can be approximated by a constant power value depending on the electrical characteristics of the used components within the USV. The second part is the mechanical part which involves the USV motion, dynamics, and environment disturbances. The mechanical power mainly depends on the thrust power; hence, the thrust force. (see Eq.(5.20)).

$$P_{USV} = P_{thrusters} + P_{static} {=} P_{Thrust} + P_{Losses} + P_{static} \tag{4.20}$$

Where, P_{USV} is the total power absorbed by the USV.

Consequently, the total consumption can be deduced as illustrated in this section from any given scenario by mathematically integrating the instantaneous power in the scenario's duration interval. In other words, the total energy consumption of the USV can be modelled by Eq.(21).

$$E_{USV} = \int P_{USV}.dt \tag{4.21}$$

Such that, P_{static} is the static power absorbed by electrical devices, $P_{thrusters}$ is the total power absorbed by the thrusters (DC motors) involving; P_{thrust} which is the useful power converted to mechanical power; P_{losses} is the power due to the power losses of the thrusters. The thrust power P_{thrust} can to be modelled as function of the environment of the boat (wind, current, speed etc.) using the USV dynamics by making the scalar product of the relative velocity vector of the USV and the thruster force given in Eq.(17).

Thus, The thrust power can be obtained as presented in Eq.(22).

$$P_{thrust} = \tau_{thrust}.V_r = M.\dot{V}_r.V_r + C(V_r).V_r.V_r + D(V_r).V_r.V_r - \tau_{wind}.V_r \qquad (4.22)$$

Thus,

$$\begin{aligned} P_{thrust} = & \begin{bmatrix} m_{11} & 0 & 0 \\ 0 & m_{22} & 0 \\ 0 & 0 & m_{33} \end{bmatrix} \cdot \begin{bmatrix} \dot{u}_r \\ \dot{v}_r \\ \dot{r}_r \end{bmatrix} \begin{bmatrix} u_r \\ v_r \\ r_r \end{bmatrix} \\ & + \begin{bmatrix} 0 & 0 & -m_{22}v_r \\ 0 & 0 & m_{11}u_r \\ m_{22}v_r & -m_{11}u_r & 0 \end{bmatrix} \cdot \begin{bmatrix} u_r \\ v_r \\ r_r \end{bmatrix} \cdot \begin{bmatrix} u_r \\ v_r \\ r_r \end{bmatrix} \\ & + \begin{bmatrix} d_{11} & 0 & 0 \\ 0 & d_{22} & 0 \\ 0 & 0 & d_{33} \end{bmatrix} \cdot \begin{bmatrix} u_r \\ v_r \\ r_r \end{bmatrix} \cdot \begin{bmatrix} u_r \\ v_r \\ r_r \end{bmatrix} - \tau_{wind} \cdot \begin{bmatrix} u_r \\ v_r \\ r_r \end{bmatrix} \end{aligned} \qquad (4.23)$$

By replacing Eq.(19) in Eq.(23), the global power model of the USVs is obtained as function of its speed and environment disturbances (water current and wind disturbances) as given in Eq.(24).

$$\begin{aligned} P_{USV} = &\begin{bmatrix} m_{11} & 0 & 0 \\ 0 & m_{22} & 0 \\ 0 & 0 & m_{33} \end{bmatrix} \cdot \begin{bmatrix} \dot{u}_r \\ \dot{v}_r \\ \dot{r}_r \end{bmatrix} \begin{bmatrix} u_r \\ v_r \\ r_r \end{bmatrix} \\ + &\begin{bmatrix} 0 & 0 & -m_{22}v_r \\ 0 & 0 & m_{11}u_r \\ m_{22}v_r & -m_{11}u_r & 0 \end{bmatrix} \cdot \begin{bmatrix} u_r \\ v_r \\ r_r \end{bmatrix} \cdot \begin{bmatrix} u_r \\ v_r \\ r_r \end{bmatrix} \\ + &\begin{bmatrix} d_{11} & 0 & 0 \\ 0 & d_{22} & 0 \\ 0 & 0 & d_{33} \end{bmatrix} \cdot \begin{bmatrix} u_r \\ v_r \\ r_r \end{bmatrix} \cdot \begin{bmatrix} u_r \\ v_r \\ r_r \end{bmatrix} \\ - &\begin{bmatrix} \frac{1}{2}\rho_a.A_{Fw}.c_x.V_w^2.\cos(\varphi - \beta_w) \\ \frac{1}{2}\rho_a.A_{Lw}.c_y.V_w^2.\sin(\varphi - \beta_w) \\ 0 \end{bmatrix} \cdot \begin{bmatrix} u_r \\ v_r \\ r_r \end{bmatrix} + P_{Losses} + P_{static} \end{aligned} \tag{4.24}$$

With:

$$V_r = \begin{bmatrix} u_r \\ v_r \\ r \end{bmatrix} = \begin{bmatrix} u - V_{cx}\cos\varphi - V_{cy}\sin\varphi \\ v - V_{cx}\sin\varphi + V_{cy}\cos\varphi \\ r \end{bmatrix}$$

Consequently, the energy consumption model is obtained by Eq.(25).

$$E_{USV} = \int P_{USV}.dt = \sum P_{USV}.\Delta t \tag{4.25}$$

In the given analytical consumption model, the terms m_{11},m_{22}, and m_{33} represent the added mass parameters; in addition, the terms d_{11},d_{22}, and d_{33} represent the dynamic coefficients according to Fossen's model [21]. Both added mass parameters and dynamic coefficients represent actually the power model parameters. The developed model is general and can be applied to any differential drive USV as long as the considered assumptions are true.

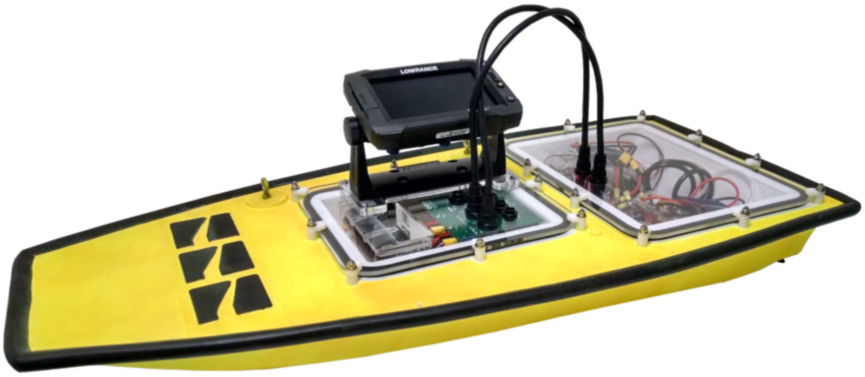


Figure 4.3: Real Lutra-prop USV

After establishing the power consumption model of the differential drive USVs as given in this chapter, the obtained model has been integrated into the powerful and the robust simulator in order to be used for various energy-based applications, the USV simulator provided in this thesis is a ROS-based software. The ROS software infrastructure and design is given in the next chapter alongside with different used simulations tools such as: Gazebo, Rviz, UWsim, USV simulator, etc. The USV simulator is used in our work for two purposes, the first one is to identify the parameters of the utra prop boat (see Figure 5.3) provided with the simulator using the reverse-engineering approach. After that, the same simulator is enriched with power management tools allowing virtual consumption estimation of USVs.

Chapter 5

Model parameters identification

5.1 Introduction

The general 3-DOF energy consumption model of the differential drive Unmanned Surface Vehicles is developed as given in the previous chapters, However, in order to exploit the developed model and to integrate it within the USV simulator, the model parameters for a particular differential drive USV available on the simulator should be identified. The identification can be done using several ways such as: analytical calculations using some predefined formulas, experimental testings, reverse engineering approach; etc.

This chapter deals with the approaches used to identify the added mass parameters m_{11},m_{22},m_{33} and the dynamic coefficient d_{11},d_{22},d_{33} of a case of study drone to obtain its full 3-DOF power model parameters expressed in in Eq.(5.24). The added mass parameters are identified based on their estimated expressions given in [13] and [40]. Whereas, the reserve engineering approach [22] is used to identify the dynamic coefficients from a recent USV simulator [1] that includes a virtual version of the case study USV. In this thesis, the considered drone is the differential drive Lutra-prop boat (see Figure 5.3) having the characteristics given in Table 6.1.

Table 5.1: Real Lutra-prop USV parameters.

Parameters	Numerical values
Length (L)	$1.06m$
Width (W)	$0.48m$
Height (H)	$0.15m$
Hull volume (V_H)	$0.02m^3$
Weight (m)	$9.7Kg$
Maximum thruster force (F_{Th})	$23N(F_{Th1} = F_{Th2} = 11.5)$
Maximum surge velocity (u)	$1.35m/s$
Moment of inertia (I_z)	$1.094Kg.m^2$
Submerged depth (D)	$0.02m$
Distance between the thrusters (d)	$0.16m$

5.2 Added mass parameters identification

The mass parameters m_{ii} include added mass contribution that represent hydraulic pressure forces and torque due to force harmonic motion of the vessel which are proportional to boat's acceleration [13]. Using Table 6.1 and the estimation of the added mass terms, the added mass parameters of the Lutra-prop differential drive boat could be obtained as follows [13, 40]:

$$m_{11} = m - X_{\dot{u}} \simeq m + 0.05 = 9.75Kg \tag{5.1}$$

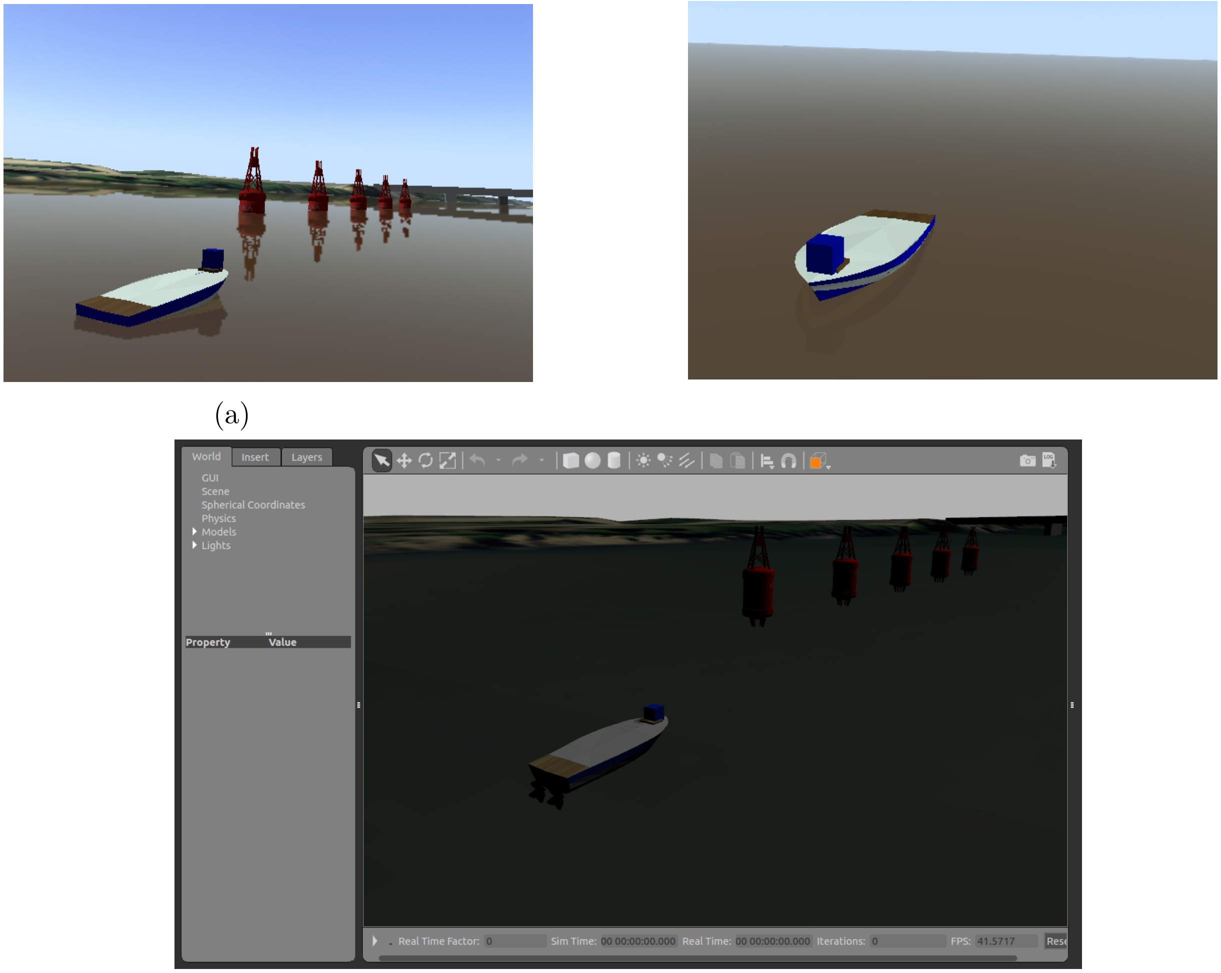
$$m_{22} = m - Y_{\dot{v}} = m + 0.5(\rho\pi D^2 L) = 10.364Kg \tag{5.2}$$

$$m_{33} = I_Z - N_{\dot{r}} = \frac{m(L^2 + W^2) + 0.5(0.1md^2 + \rho\pi D^2 L^3)}{12} \simeq 1.158Kg \tag{5.3}$$

where ρ is the water density ($\approx 1025Kg/m^3$ for salt water at moderate temperature)

5.3 Dynamic coefficients identification

The USV simulator [1] is used to identify the hydrodynamic coefficient parameters d_{ii} of the virtual Lutra-prop differential drive USV (Figure 6.1) using the reverse-engineering approach. In this section, three considered scenarios are used to identify each parameter: Scenario (a) (linear motion) which allows us to identify the parameter d_{11} in the steady state phase using Eq.(5.9). Scenario (b) (circular motion) which provides the identification of d_{22} using Eq.(5.10) in the steady state mode. And finally Scenario (c) (rotation around the center point) which enables the identification of d_{33} in the steady state mode as well using Eq.(5.11).



Real Time Factor: 0 Sim Time: 00 00:00:00.000 Real Time: 00 00:00:00.000 Iterations: 0 FPS: 41.5717 Rese

(a)

(b)

Figure 5.1: Lutra-prop boat on the under water simulator window (a) and on Gazebo window (b)

Since the USV simulator is a ROS-based software, a ROS node named */plot* is implemented to subscribe to */state* topic to repeatedly receive the odometry information of the boat as odometry message type involving the position of the boat in $\{e\}$ frame and its velocity vector in $\{b\}$ frame. The odometry information is published by Gazebo ROS node via position and velocity sensors provided within the

Gazebo simulator. The ROS node */plot* plots different graphs needed to identify each parameter from each scenario as described in this section.

Scenario (a) - Linear motion

The USV is configured to move longitudinally with maximum thrust $F_{Th1} = F_{Th2} = 11.5N$) by neglecting all disturbance. The USV moves through a linear trajectory given in Figure 6.2. The surge velocity for this scenario was recorded and plotted by the ROS node process */plot* and given in Figure 6.3. In the steady state phase of scenario (a), the sway velocity and yaw rate are null, as well as the surge acceleration. i.e. $\dot{u} = r = v = 0$, and the boat moves at the maximum speed i.e. $u = u_{max} = 1.35m/s$. Thus, by using Eq.(5.9) the first dynamic coefficient d_{11} is obtained as given in Eq.(6.4).

$$d_{11} = X_u = \frac{F_{Th}}{u_{\max}} \simeq 16.296Ns/m \tag{5.4}$$

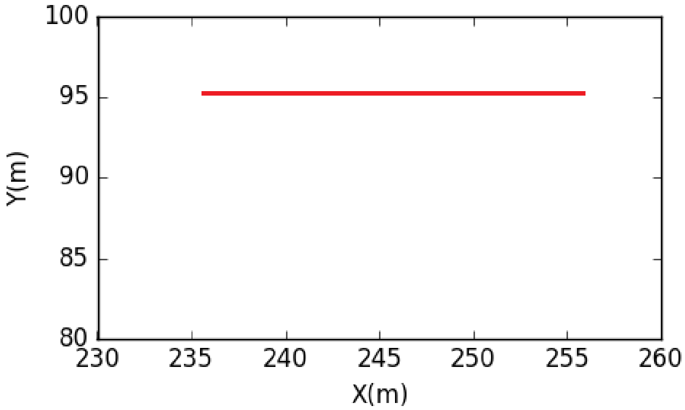


Figure 5.2: Trajectory line of scenario a

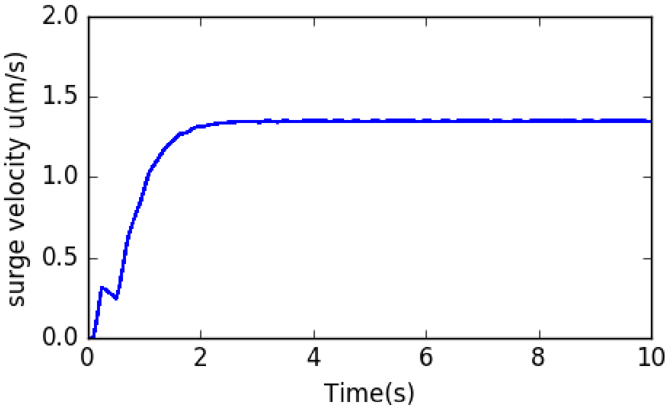


Figure 5.3: Surge velocity $u(t)$ representation for scenario a

Scenario (b) - Circular motion

The USV is configured in this scenario to be controlled only by one thruster at maximum thrust value i.e. $F_{Th1} = 11.5N$, $F_{Th2} = 0N$. The boat follows a circular trajectory shown in Figure 6.4. The instantaneous values of surge, sway, and yaw rate u,v,r are recorded through this scenario and plotted by the same ROS node process and presented in Figure 6.5 to Figure 6.7. In the steady state mode, the sway acceleration is a null value i.e. $\dot{v} = 0$. Thus, by using Eq.(5.10), we obtain the d_{22} parameter as given in Eq.(6.5).

$$d_{22} = Y_v = \frac{-m_{11}ur}{v} = \frac{-9.75 \times 0.92 \times (-0.25)}{0.22} \simeq 10.193 Kg.rad/s \tag{5.5}$$

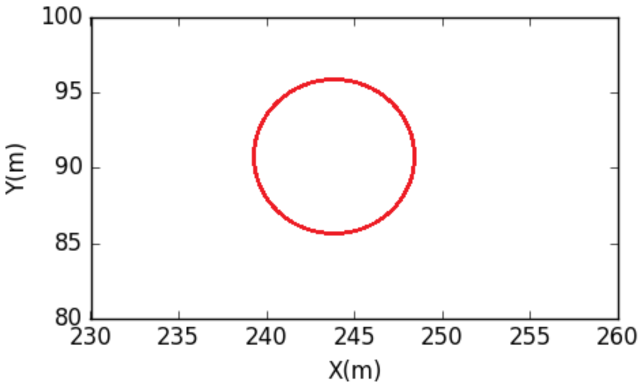


Figure 5.4: The circular trajectory taken by the USV for scenario b

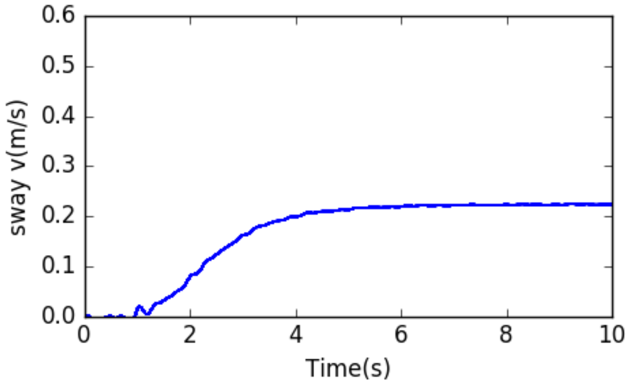


Figure 5.5: Sway velocity $v(t)$ for scenario b

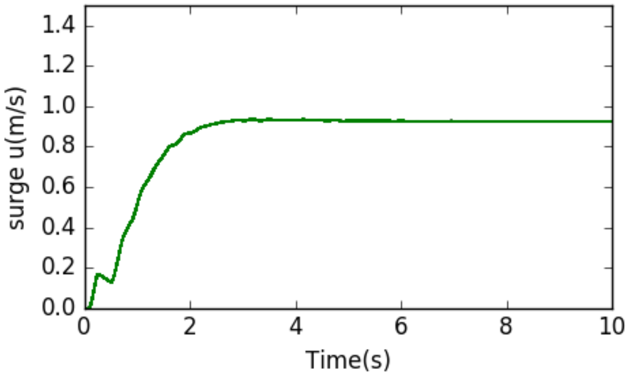


Figure 5.6: Surge velocity $u(t)$ for scenario b

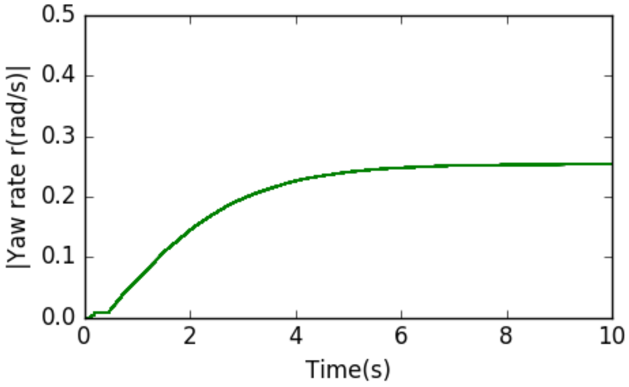


Figure 5.7: Yaw rate $r(t)$ for scenario b

Scenario (c)- Rotation around about the vertical axis

In this scenario, the USV thrusters were configured to thrust in opposite directions i.e. $F_{Th1} = -F_{Th2} = 11.5N$ making the USV rotating around the vertical axis as illustrated in Figure 6.8. The yaw rate was recorded and plotted by the process of the ROS node */plot* and given in Figure 6.9. In the steady state phase of this scenario, the surge and sway velocities are null values as well as the yaw rate acceleration i.e. $\dot{r} = u = v = 0$, which implies that the d_{33} coefficient can be obtained from Eq.(5.11) as follows:

$$d_{33} = N_r = \frac{F_N.d}{2r_{\max}} \simeq 4.63Nms/rad \tag{5.6}$$

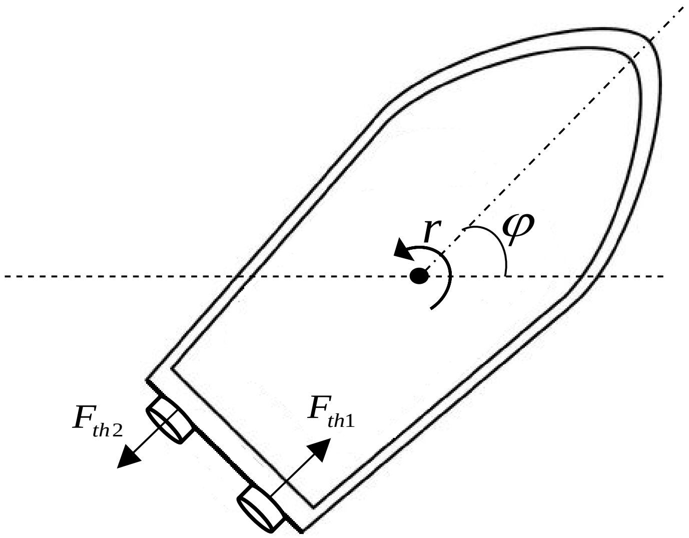


Figure 5.8: Thrust force representation for scenario c

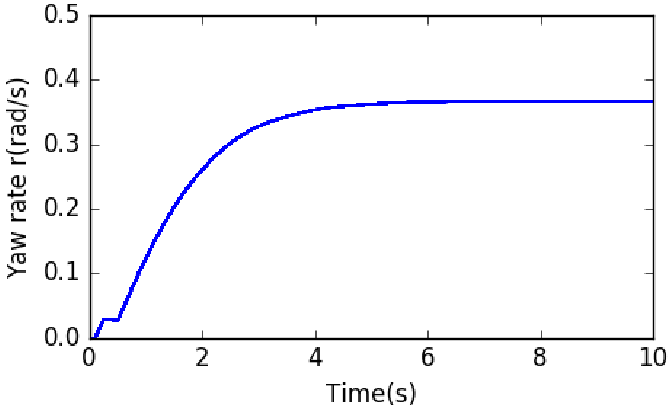


Figure 5.9: Yaw rate $r(t)$ representation for scenario c

5.3.1 Power model expression for the case of study USV

The power model expression of the Lutra-prop differential drive USV can be finally expressed after identifying its parameters. The complete 3-DOF dynamic model parameters are summarized in Table 6.2 including the added mass parameters and the dynamic coefficients. The obtained results have been validated through the same simulator by verifying Eq.(5.9), Eq.(5.10), and Eq.(5.11) which indicate the correctness of obtained parameters and the effectiveness of the proposed identification approaches.

Table 5.2: Lutra-prop boat dynamic model parameters

Parameter	Numerical value
$d_{11} = -X_u$	$16.296 Ns/m$
$X_{\dot{u}}$	$-0.050 Kg$
$d_{22} = -Y_v$	$10.193 Kgrad/s$
$Y_{\dot{v}}$	$-0.664 Kg$
$d_{33} = -N_r$	$4.630 Nms/rad$
$N_{\dot{r}}$	$-0.064 Kg.m^2$
m_{11}	$9.750 Kg$
m_{22}	$10.364 Kg$
m_{33}	$1.158 Kg.m^2$

Moreover, the USV simulator is provided with Gazebo model plug-in that applies both lateral and frontal wind forces on the centre point of the USV as a function of wind velocity and some constant parameters given in Eq.(5.19) such as the lateral and frontal areas exposed to the wind. In addition to the dynamic model parameters of the Lutra-prop boat given in Table 6.2, the wind force parameters used by the plug-in are extracted and given in Table 6.3.

Table 5.3: Wind force parameters

Parameter	Numerical value
ρ_a	$1.184 Kg/m^3$
A_{Fw}	$0.08m^2$
A_{Lw}	$0.18m^2$
c_x	0.68
c_y	1.11

Finally, the Lutra-prop differential boat USV energy consumption model is established as function of the environment and given in Eq.(6.7). As a consequence, Coriolis centripetal matrix has been omitted by calculation; thus, it has no effect in the energy consumption of the surface drones.

$$\begin{aligned} P_{USV} = & \begin{bmatrix} 6.78 & 0 & 0 \\ 0 & 1.06 & 0 \\ 0 & 0 & 6.03 \end{bmatrix} \cdot \begin{bmatrix} \dot{u}_r \\ \dot{v}_r \\ \dot{r}_r \end{bmatrix} \begin{bmatrix} u_r \\ v_r \\ r_r \end{bmatrix} \\ & + \begin{bmatrix} 16.30 & 0 & 0 \\ 0 & 7.09 & 0 \\ 0 & 0 & 4.63 \end{bmatrix} \cdot \begin{bmatrix} u_r \\ v_r \\ r_r \end{bmatrix} \cdot \begin{bmatrix} u_r \\ v_r \\ r_r \end{bmatrix} \\ & - \begin{bmatrix} 0.032 V_w^2 . \cos(\varphi - \beta_w) \\ 0.105 V_w^2 . \sin(\varphi - \beta_w) \\ 0 \end{bmatrix} \cdot \begin{bmatrix} u_r \\ v_r \\ r_r \end{bmatrix} + P_{static} \end{aligned} \tag{5.7}$$

With:

$$\begin{bmatrix} u_r \\ v_r \\ r \end{bmatrix} = \begin{bmatrix} u - V_{cx}\cos\varphi - V_{cy}\sin\varphi \\ v - V_{cx}\sin\varphi + V_{cy}\cos\varphi \\ r \end{bmatrix} \tag{5.8}$$

Where: $[u_r,v_r,r]^T$ is the USV's relative velocity vector, $[u,v,r]^T$ is the actual USV velocity vector, $[V_{cx},V_{cy}]^T$ is the water current speed vector given in $\{e\}$ frame, $[V_w,\beta_w]^T$ is the polar vector of the wind speed in $\{e\}$ frame as well. φ is the angle of attack of the USV. As well as P_{static} is the static power absorbed by the electrical components the USV.

The energy consumption model of the differential drive Lutra-boat is well established and presented in this chapter. However, in order to verify and test the obtained model, we propose again to integrate it into the simulator engine by implementing additional processes that calculate instantaneously the power absorbed by the USV when operating. Therefore, the simulator is well enriched with power management and estimation tools that can be used further to solve several energy-based applications such as path-planing, mission management, tasks scheduling, etc.

Besides, the approaches used to enrich the simulator with power management tools are well illustrated in the next chapter. Furthermore, in order to visualize the obtained results, several scenarios are proposed in the next chapter as well, where the USV speed and consumption variation were recorded and presented as the simulation results. .

Chapter 6

Simulation and results

6.1 Introduction

As presented in the previous chapters, the analytical energy model of the differential drive USV has been established as a function of its velocity vector and environment disturbances. Moreover, the parameters of the developed model have been identified through several approaches. Consequently, the obtained model can be integrated into the same presented USV simulator to verify and to approve its correctness and usefulness.

This chapter illustrates the approach used to enrich the USV simulator with power and energy management tools to instantaneously calculate and to monitor the real-time USV's dynamic power consumption. As already reported, the used simulator is an open-source ROS/Gazebo-based software that models different virtual boats with realistic behaviours and environment disturbances [1]; therefore, it uses ROS processes (known as ROS nodes) for navigation and control.

6.2 Power model integration into USV simulator

The power model integration approach can be achieved by implementing custom ROS nodes that instantaneously calculate the real-time USV consumption. In addition, ROS nodes can communicate with each other by sending and receiving messages organized into specific categories named topics [52]. Moreover, the interconnected ROS nodes can be represented using a directed graph (see Figure 7.1) such that the vertices represent the active ROS nodes (or processes) while the edges represent the implemented topics.

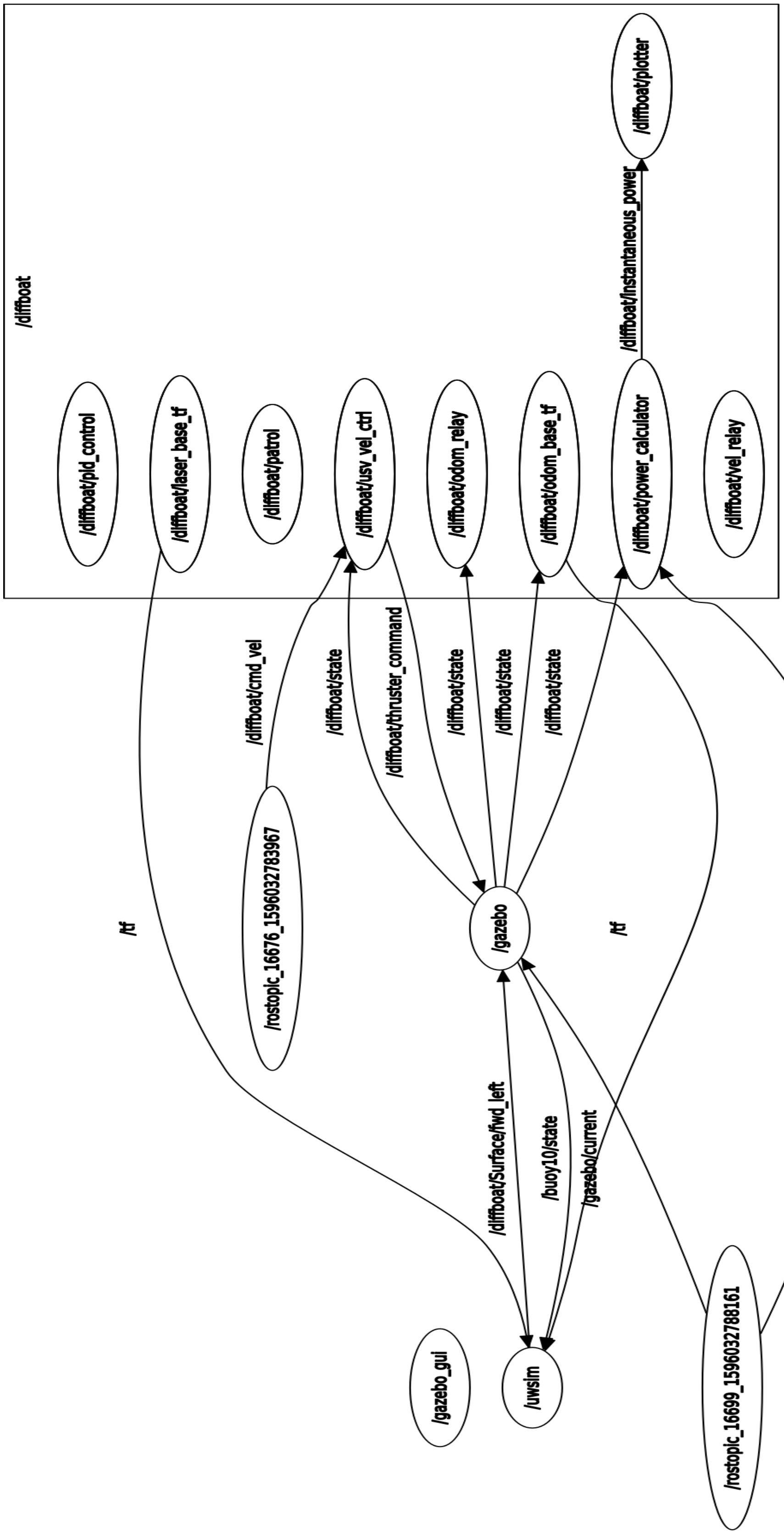


Figure 6.1: The RQT-graph showing the interconnected ROS processes

The given graph is also known as the RQT-graph obtained using the *rqt_graph* command in the Linux terminal window [52]. By convention, when a ROS node sends messages to a given topic, means that the node is publishing to that topic. Alternatively, when a ROS node receives messages via a given topic, means that the node is subscribing to that topic.

Consequently, we notice from Figure 7.1 that when launching a particular differential boat scenario, nine (9) ROS nodes including the implemented power nodes start (highlighted by a square). For instance, the ROS node named */usv_vel_ctrl* subscribes to the topic named */cmd_vel* to receive the desired velocity; thereafter, it publishes the thruster command to control the boat's propellers. Similarly, the */power_calculator* node subscribes to two different topics, the first one is named */state* used to get the boat's odometry information (position and velocity vectors), while the second topic is named */current* used to obtain the published water current speed; thereafter, the ROS node calculate the real-time power and publishes its value to the */instantaneous_power* topic to be plotted by the ROS node */plotter*.

Besides, the power calculation process */power_calculator* is well illustrated in the flowchart given in Figure 7.2. The process starts be creating the ROS node say: */power_calculator* that subscribes to */diffboat/state* topic to receive odometry information involving the actual USV's velocity components and position coordinates according to the capabilities of Gazebo and its available sensors. The received information is used to calculate the USV's relative speed vector components and accelerations. In addition to the odometry information, the node repeatedly subscribes to the */gazebo/current* topic to read the water current status. The ROS node uses this data in addition to a configurable wind speed to calculate instantaneously the absorbed dynamic power by the USV during any given scenario based on Eq.(6.7). Finally, the process loop ends by publishing the calculated power and starts over again until the running scenario is finished.

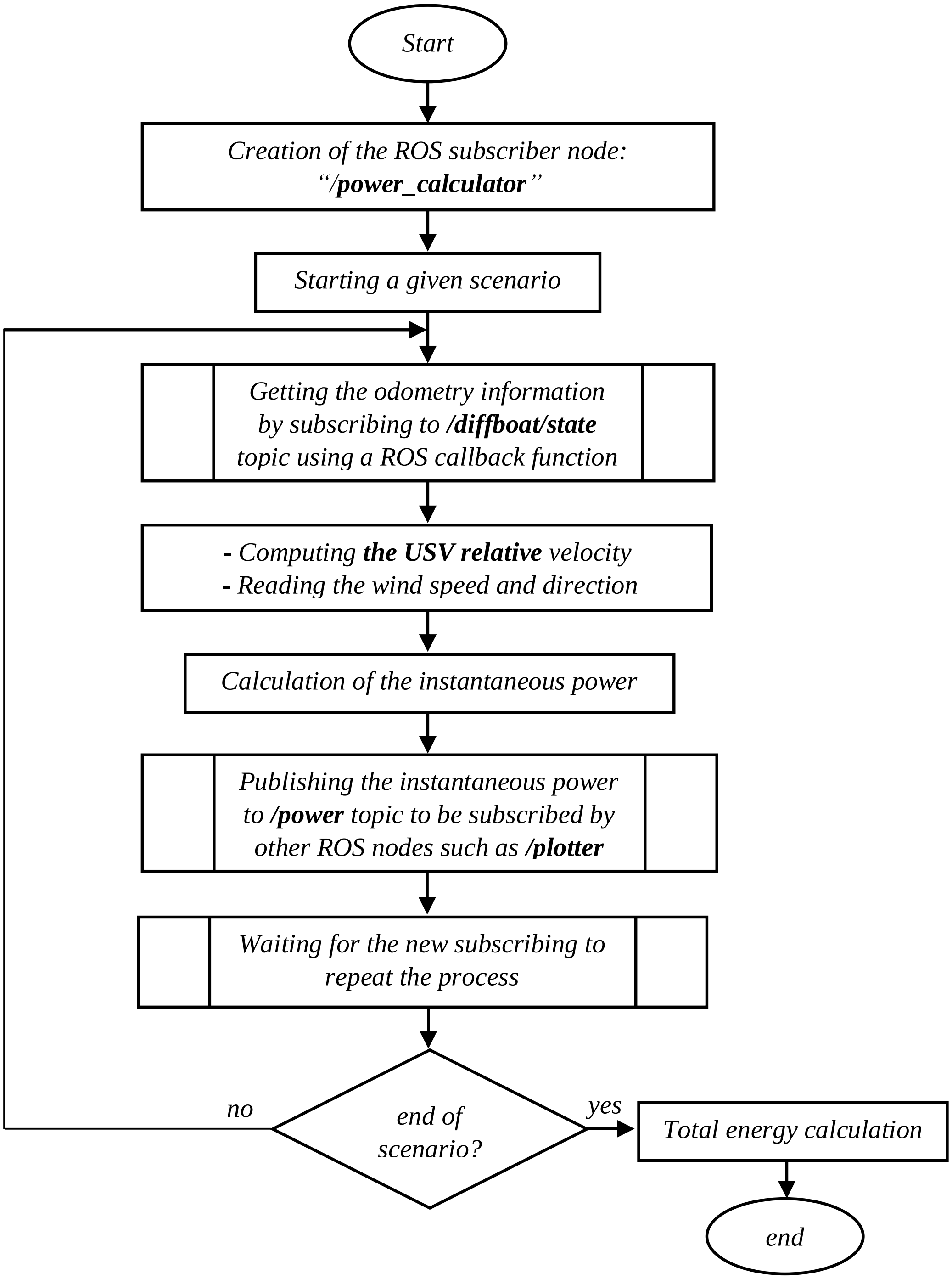

Start
Creation of the ROS subscriber node:
"/power_calculator"
Starting a given scenario
Getting the odometry information
by subscribing to /diffboat/state
topic using a ROS callback function
- Computing the USV relative velocity
- Reading the wind speed and direction
Calculation of the instantaneous power
Publishing the instantaneous power
to /power topic to be subscribed by
other ROS nodes such as /plotter
Waiting for the new subscribing to
repeat the process
end of
scenario?
no
yes
Total energy calculation
end

Figure 6.2: Flowchart of the power calculation process

6.3 Simulation and Results

In order to represent the effectiveness and usefulness of the energy modelling, identification, and integration into the simulation environment approaches, several scenarios are conducted with and without environment disturbances (mainly: wind and water current). In this section, we go through seven (7) different and independent executed scenarios. The surge velocity $u(t)$ and the real-time dynamic power absorbed by the USV are instantaneously calculated and recorded for each scenario by neglecting the static power. These scenarios have been chosen to apply the established model in such a way to be as realistic as possible to real situation that can happen, in such a manner that the wind and current disturbances are not considered for scenarios 1 and 5, different water current speeds and directions are considered for scenarios 2, 3, 4, 6, and 7; furthermore, the wind disturbance is considered for scenarios 6 and 7. More details for each scenario is well described in this section. Moreover, For each scenario, the implemented ROS processes calculate and return instantaneously the absorbed power based on the po wer model given in Eq.(6.7) and on the published odometry data as previously explained. Besides, we present at the end of this section the effect of a given USV speed on its power consumption.

6.3.1 Scenario # 1

In this scenario, the boat is configured to move from a starting point $(240; 95)$ to a target point $(270; 965)$ represented in $\{e\}$ reference frame at a maximum speed ($u = 1.34 m/s$) without any disturbances (see Figure 7.3). The surge velocity $u(t)$ and the calculated power are recorded and plotted as given in Figure 7.4 and Figure 7.5 respectively. The total consumption returned by the implemented power processes at the end of the scenario using Eq.(5.25) and Eq.(6.7) is $607.87J$.

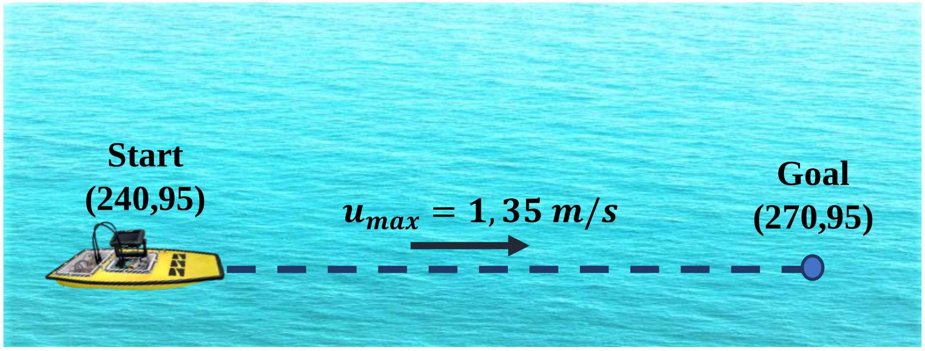


Figure 6.3: Scenario 1 representation

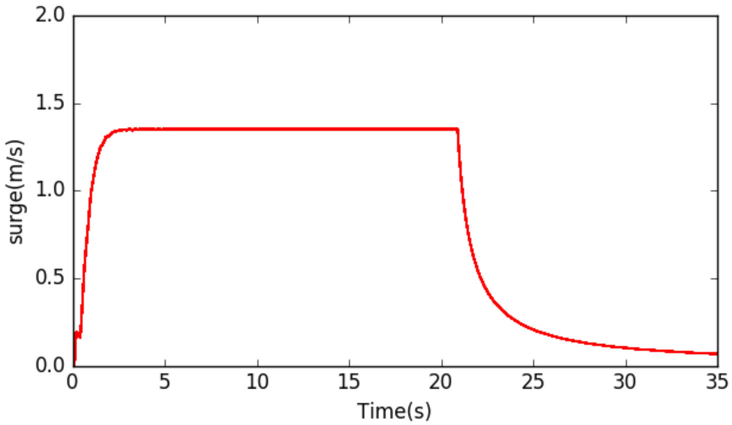


Figure 6.4: Surge velocity $u(t)$ recorded for scenario 1

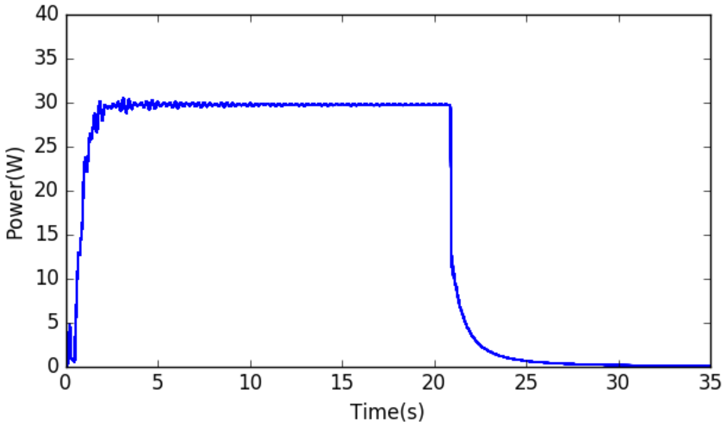


Figure 6.5: Power consumption $p(t)$ variation for scenario 1

6.3.2 Scenario # 2

The second scenario is similar to the first one except that the water current is configured now to run in the same direction of the drone's motion at $0.40m/s$ (see Figure 7.6). The recorded surge velocity and power consumption variations are given in Figure 7.7 and Figure 7.8. From these figures we notice that the second scenario is faster than the first one which took only 16.5 seconds to finish and consumed about $461.81J$ only. The drone's willingness to stay at the target point resisting the water current flux results in additional power consumption after the end of the scenario.

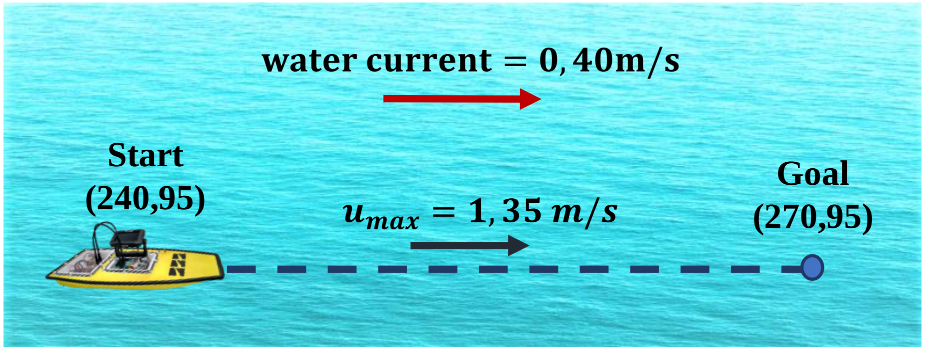


Figure 6.6: Scenario 2 representation

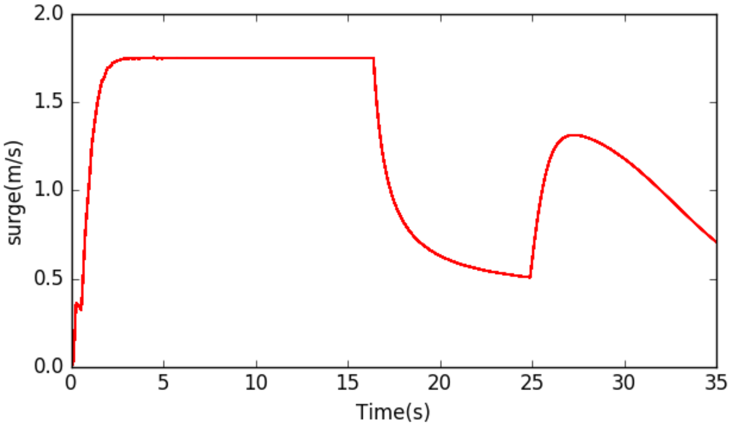


Figure 6.7: Surge velocity $u(t)$ recorded for scenario 2

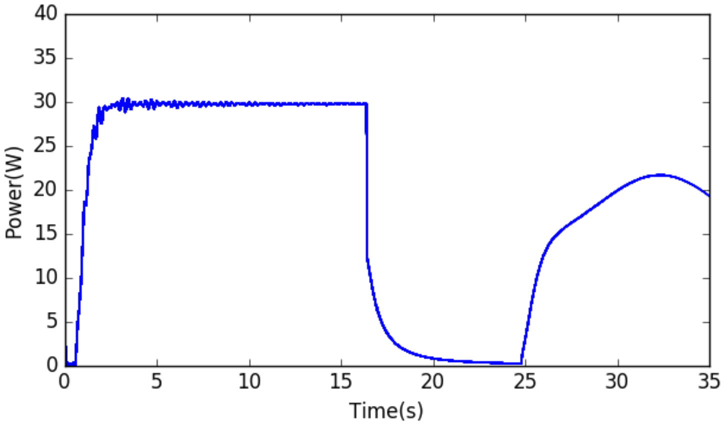


Figure 6.8: Power consumption $p(t)$ variation for scenario 2

6.3.3 Scenario # 3

As given in Figure 7.9, the third scenario is configured such that the water flows in the direction of 45° at a speed of $0.40m/s$. The surge velocity of the boat and its power consumption are plotted and represented in Figure 7.10 and Figure 7.11. We notice that the third scenario consumes much power than the previous one, and this is due to the water current flow direction constraint. The total energy consumed at the end if this scenario was $573.78J$.

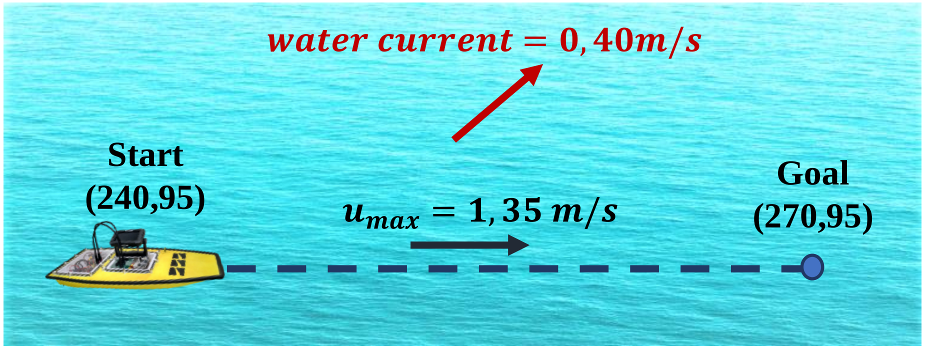


Figure 6.9: Scenario 3 representation

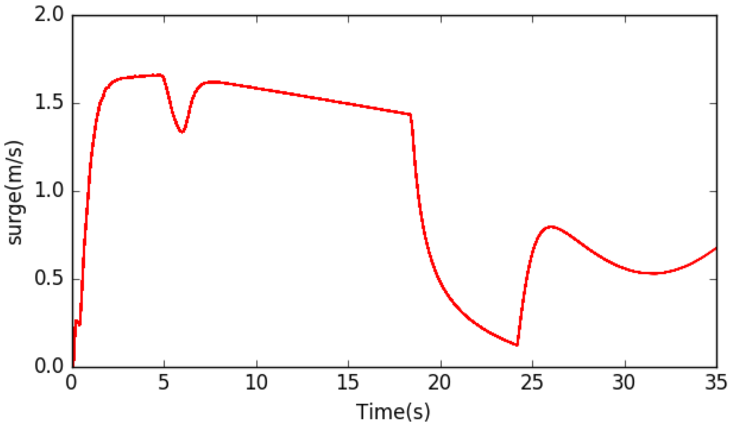


Figure 6.10: Surge velocity $u(t)$ recorded for scenario 3

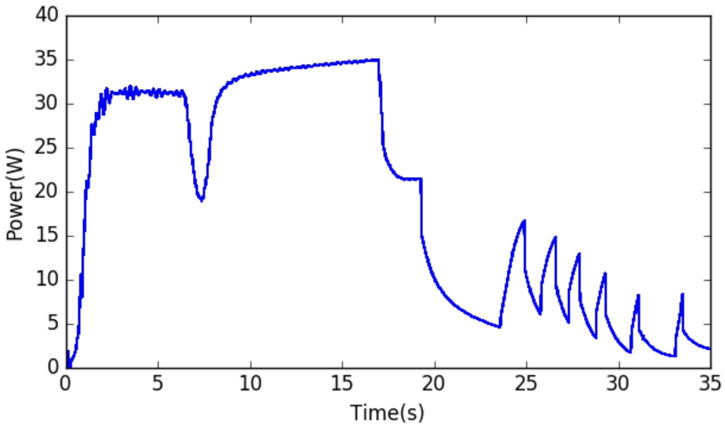


Figure 6.11: Power consumption $p(t)$ variation for scenario 3

6.3.4 Scenario # 4

This scenario is similar to the third one except that the drone comes back from the target point to its original point (see Figure 7.12). The surge velocity component and power variation are represented in Figure 7.13 and Figure 7.14 respectively. We notice that this scenario is longer than the previous one and consumes much power ($825.49J$). We observe also in Figure 7.13 that the drone's velocity can not reach its maximum value since it is moving against the water flow.

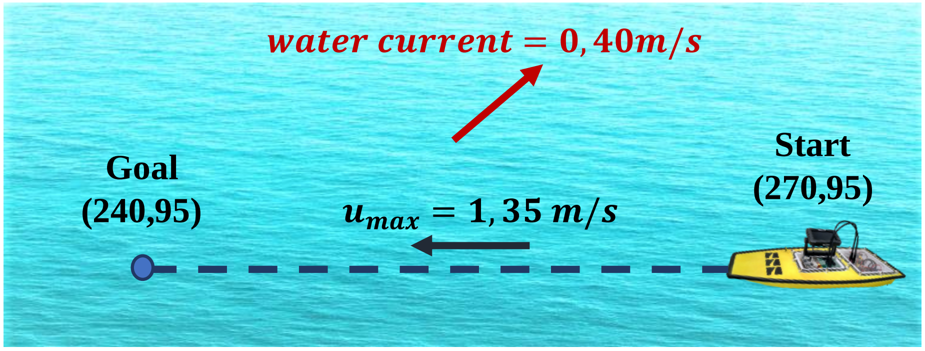


Figure 6.12: Scenario 4 representation

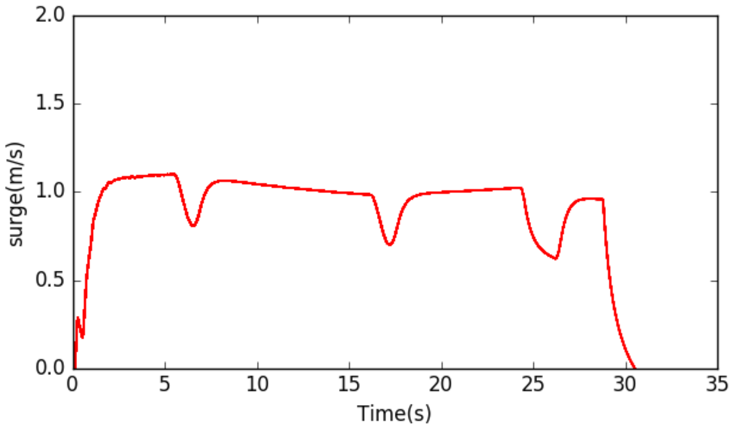


Figure 6.13: Surge velocity $u(t)$ recorded for scenario 4

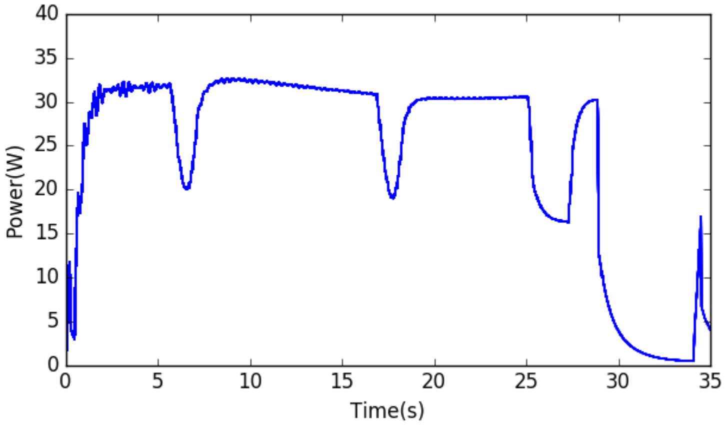


Figure 6.14: Power consumption $p(t)$ variation for scenario 4

6.3.5 Scenario # 5

The boat is configured in this scenario to move from the starting point to the target point at a maximum speed of $1m/s$ (see Figure 7.15). The surge velocity and power consumption variations are plotted and represented in Figure 7.16 and Figure 7.17. The total energy required to complete the scenario is returned by the power process which equals to $460.47J$. The overshoots shown in the two graphs are obtained due to the default controller used within the USV simulator which demonstrates that the optimal controller design plays a significant impact on the power consumption; thus, our power simulator is a very useful tool to design and tune the controller.

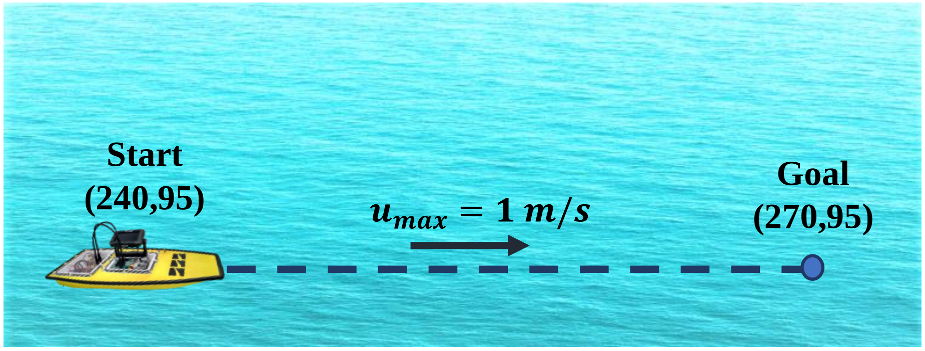


Figure 6.15: Scenario 5 representation

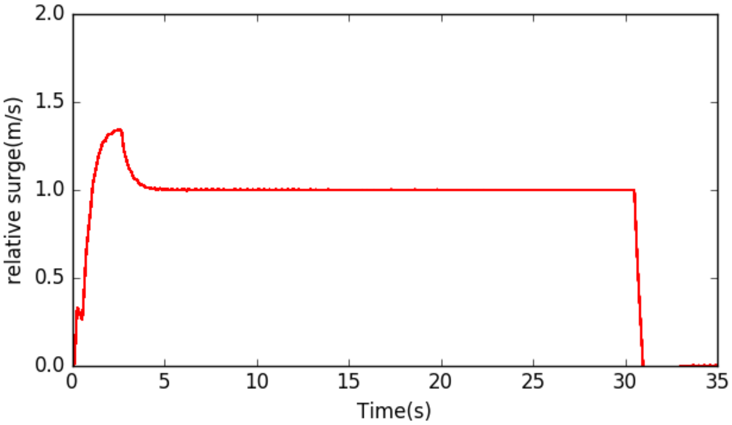


Figure 6.16: Relative surge velocity $u(t)$ recorded for scenario 5

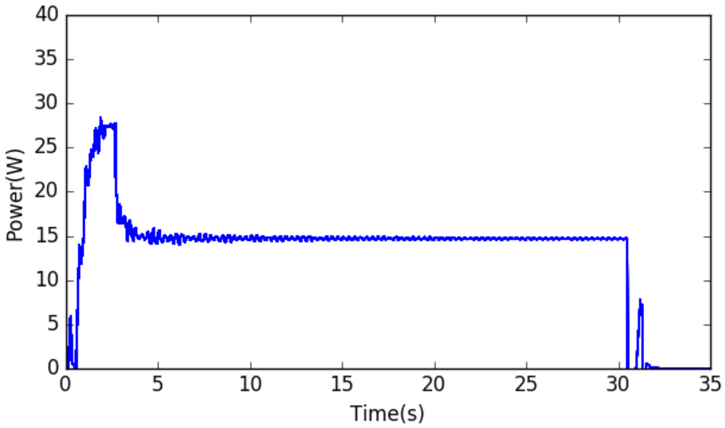


Figure 6.17: Power consumption $p(t)$ variation for scenario 5

6.3.6 Scenario # 6

In this scenario, the vehicle is configured to move from the starting point to the target point at a maximum speed of $1m/s$ but with the presence of water current disturbance ($0.30m/s$,45°) and wind disturbance ($0.20m/s$,30°) as represented in Figure 7.18. The relative velocity given in Figure 7.19 is smaller than the USV's speed because the disturbances are not absolutely obstructing the boat's motion. The power variation is illustrated in Figure 7.19 and the total consumption of this scenarios given by the implemented power process is about $305.73J$.

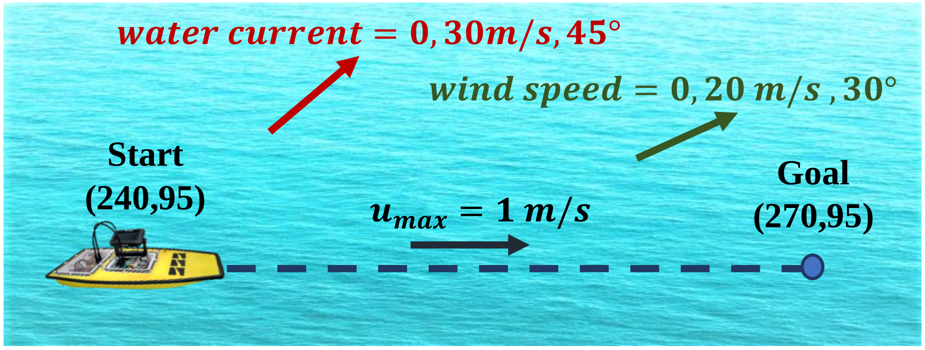


Figure 6.18: Scenario 6 representation

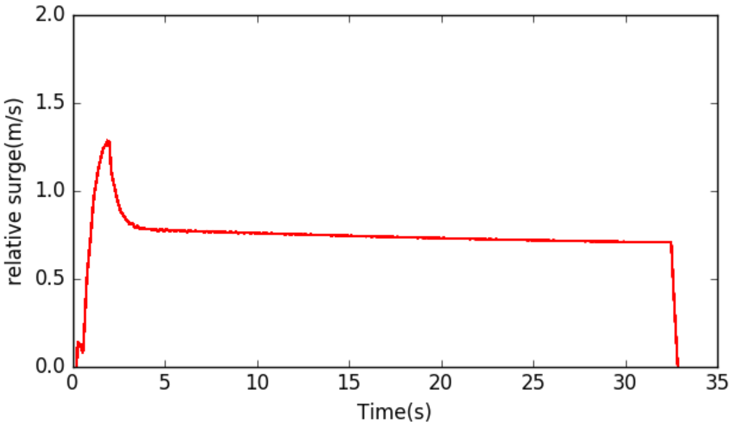


Figure 6.19: Relative surge velocity $u(t)$ recorded for scenario 6

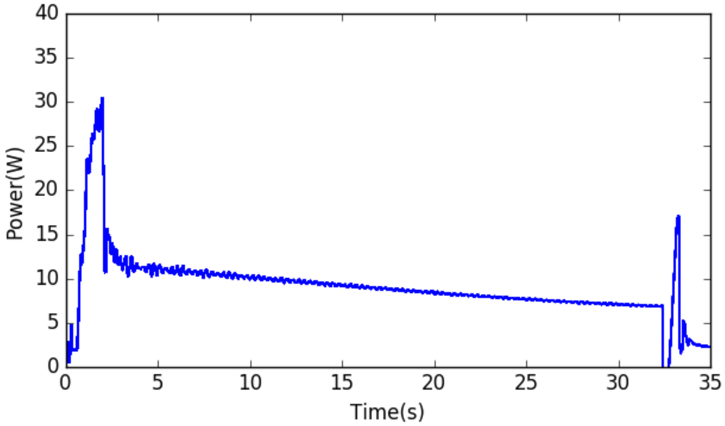


Figure 6.20: Power consumption $p(t)$ variation for scenario 6

6.3.7 Scenario # 7

This scenario is similar to scenario 6 except that the drone should come back from the target point to the original one under the same disturbances (see Figure 7.21). From Figure 22 and Figure 7.23, we notice that this scenario consumes much power than the previous one since the disturbances are obstructing the drone's motion; thus, the total energy required to complete this scenario is evaluated by the power process which equals to $898.44J$.

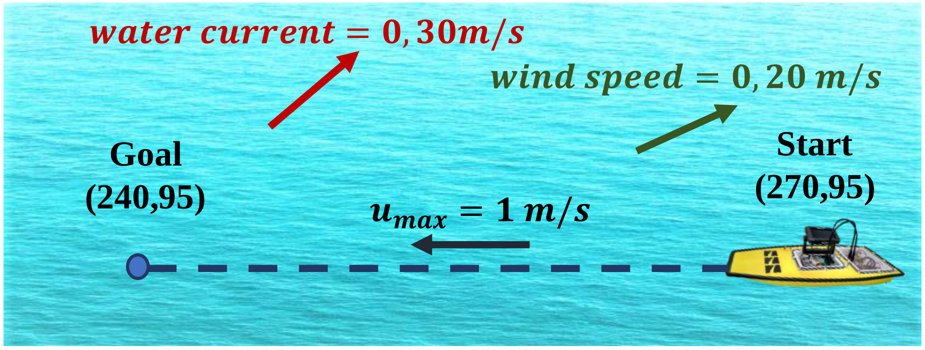


Figure 6.21: Scenario 7 representation

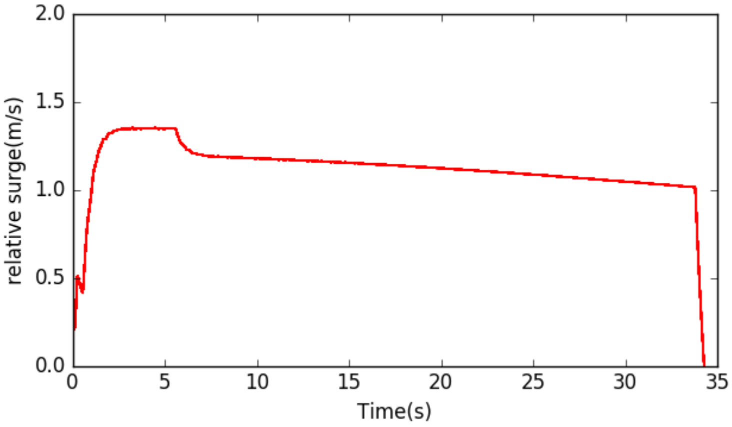


Figure 6.22: Relative surge velocity $u(t)$ recorded for scenario 7

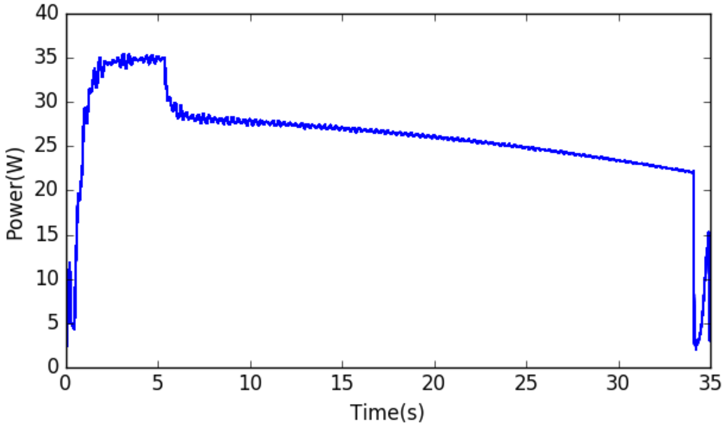


Figure 6.23: Power consumption $p(t)$ variation for scenario 7

As a result, The obtained duration and consumption values for each of the above described scenarios are summarized in Table 7.1. By comparing the first four scenarios, we deduce that the drone took different duration with different consumption due the disturbances variations. Furthermore, by comparing scenario 1 and scenario 5, we observe that the velocity variation of the surface drones has a significant effect of its consumption. This effect is well introduced in the next sub section.

Table 6.1: Results summary

Scenario number	Total time (s)	Total energy ($joules$)
1	21.0	607.87
2	16.5	441.81
3	18.5	573.78
4	29.0	825.49
5	30.5	460.47
6	32.5	305.73
7	33.5	898.44

6.3.8 The effect of the USV speed on the energy behaviour

This section shows the effect of the USV speed on its power consummation. The virtual USV was configured to complete different scenarios such that, in each scenario the USV is supposed to move from one point to another at a given linear speed by skipping wind and water current disturbances. After that, the steady state power consumption for each scenario is calculated by the implemented power processes and recorded in the graph of Figure 7.24 which illustrates the absorbed power versus the surge velocity variation given by the simulator. The given graph indicates that the power consumption varies exponentially with respect to the USV's speed in the case study speed interval. As already hypothesized in chapter 3, the non-linear drag coefficients should not be ignored if the surface drone speed exceeds a given value (mainly $> 1.5m/s$) and obviously this will lead to a more exponential power and energy variation.

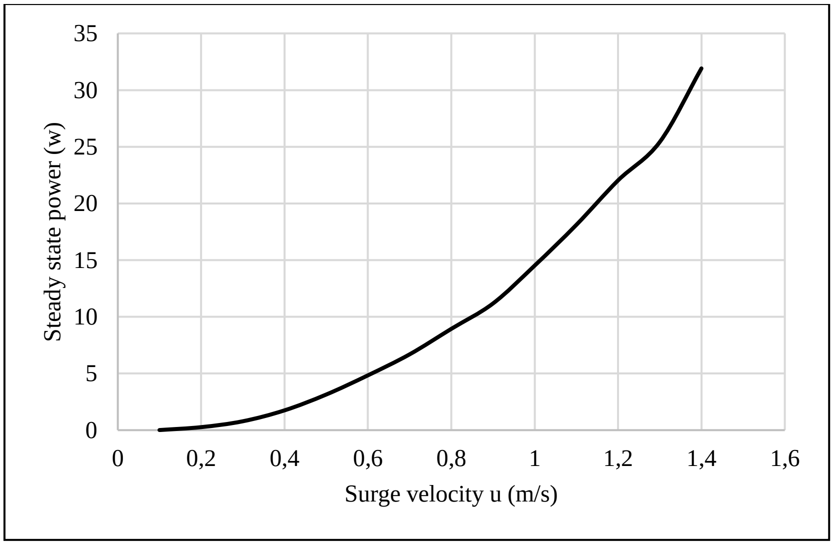


Figure 6.24: The effect of the linear velocity on the power consumption

As a consequence, the energy behaviour of the differential drive USV is verified through seven different realistic scenarios where the real-time power variation was calculated and recorded. As well as the effect of the USV speed on its consumption behaviour is given in this chapter. In other words, the obtained results can be hybridised and generalized to more complicated scenarios numerically without any hardware setup or requirement. .

Conclusion

The energy consumption model of the Unmanned Surface Vehicles was established first and presented as a function of their velocity vector components and environment disturbances (wind, water current, etc.) based on the Three-degrees-of-freedom (3-DOF) dynamic model of surface vessels. The power model is presented as a matrix form including the dynamic behaviour parameters in addition to a set of wind and water current disturbances' coefficients.

A reverse-engineering approach was applied on a recent and very robust ROS and Gazebo-based USV simulation environment that realistically simulate USV behaviours in realistic environments to identify the dynamic model parameters of the Lutra-prop USV; thus, its complete energy consumption model is deduced. The wind dynamic parameters were obtained from the Gazebo's fluid dynamics plug-in integrated within the simulator package. Thereafter, the energy consumption model was well integrated into the simulation environment and verified through seven different scenarios; thus, the simulator is enriched with power management and estimation tools. The provided results for the seven scenarios show the variation of the USV consumption as environment conditions change and allow us also to evaluate the relationship between the USV's speed and its consumption.

The presented work shows the importance of simulator-based power management and estimation which plays a significant role to resolve many energy-based problems such as: optimal path planning investigations, optimal tasks scheduling, autonomy problems, optimal control systems design, etc. to be considered in our main further work as indicated in our research paper [54].

The presented work motivates us to think about identifying more realistic consumption models by considering the wave effect and the non-linear drag impact to be reported in further papers in order to get a more detail led and accurate power model and eventually, this approach can be further be confirmed by carrying out experimental studies. In addition; the presented work can be exploited with different unmanned vehicle types such as aerial drones or under water drones by integrating their power models into their corresponding simulators such as those presented in [49, 50, 51] after establishing their analytical power consumption models as function of their environment and its disturbances.

References

[1] M. Paravisi, D. H Santos, V. Jorge, G. Heck, L. M. Gonçalves et A. Amory, 'Unmanned surface vehicle simulator with realistic environmental disturbances', *Sensors*, t. 19, nº 5, p. 1068, 2019 (cf. p. iv, vii, 5, 6, 17, 25, 26, 32-35, 37, 56, 57, 66).

[2] J. Brown, C. Tuggle, J. MacMahan et A. Reniers, 'The use of autonomous vehicles for spatially measuring mean velocity profiles in rivers and estuaries', *Intelligent Service Robotics*, t. 4, nº 4, p. 233, 2011 (cf. p. 1, 2).

[3] J. Villa, J. Paez, C. Quintero, E. Yime et J. Cabrera, 'Design and control of an unmanned surface vehicle for environmental monitoring applications', in *2016 IEEE Colombian Conference on Robotics and Automation (CCRA)*, IEEE, 2016, p. 1-5 (cf. p. 1).

[4] R. R. Murphy, E. Steimle, M. Hall, M. Lindemuth, D. Trejo, S. Hurlebaus, Z. Medina-Cetina et D. Slocum, 'Robot-assisted bridge inspection', *Journal of Intelligent & Robotic Systems*, t. 64, nº 1, p. 77-95, 2011 (cf. p. 1, 2).

[5] S. Campbell, W. Naeem et G. W. Irwin, 'A review on improving the autonomy of unmanned surface vehicles through intelligent collision avoidance manoeuvres', *Annual Reviews in Control*, t. 36, nº 2, p. 267-283, 2012 (cf. p. 2).

[6] R.-j. Yan, S. Pang, H.-b. Sun et Y.-j. Pang, 'Development and missions of unmanned surface vehicle', *Journal of Marine Science and Application*, t. 9, nº 4, p. 451-457, 2010 (cf. p. 2).

[7] S. Brizzolara, T. Curtin, M. Bovio et G. Vernengo, 'Concept design and hydrodynamic optimization of an innovative swath usv by cfd methods', *Ocean dynamics*, t. 62, nº 2, p. 227-237, 2012 (cf. p. 2).

[8] V. A. Jorge, R. Granada, R. G. Maidana, D. A. Jurak, G. Heck, A. P. Negreiros, D. H. dos Santos, L. M. Gonçalves et A. M. Amory, 'A survey on unmanned surface vehicles for disaster robotics: Main challenges and directions', *Sensors*, t. 19, nº 3, p. 702, 2019 (cf. p. 3).

[9] H. Niu, Y. Lu, A. Savvaris et A. Tsourdos, 'An energy-efficient path planning algorithm for unmanned surface vehicles', *Ocean Engineering*, t. 161, p. 308-321, 2018 (cf. p. 3, 5, 15).

[10] C. Yoon, S. Lee, Y. Choi, R. Ha et H. Cha, 'Accurate power modeling of modern mobile application processors', *Journal of Systems Architecture*, t. 81, p. 17-31, 2017 (cf. p. 3, 14).

[11] N. Khare et P. Singh, 'Modeling and optimization of a hybrid power system for an unmanned surface vehicle', *Journal of Power Sources*, t. 198, p. 368-377, 2012 (cf. p. 3).

[12] B. Moyer, 'Low-power design for embedded processors', *Proceedings of the IEEE*, t. 89, n° 11, p. 1576-1587, 2001 (cf. p. 4).

[13] K. R. Muske, H. Ashrafiuon, G. Haas, R. McCloskey et T. Flynn, 'Identification of a control oriented nonlinear dynamic usv model', in *2008 American Control Conference*, IEEE, 2008, p. 562-567 (cf. p. 4, 15, 56, 57).

[14] P. Asgharian et Z. H. Azizul, 'Proposed efficient design for unmanned surface vehicles', *arXiv preprint arXiv:2009.01284*, 2020 (cf. p. 4).

[15] L. Zhang, J. Kim et J. Sun, 'Energy modeling and experimental validation of four-wheel mecanum mobile robots for energy-optimal motion control', *Symmetry*, t. 11, n° 11, p. 1372, 2019 (cf. p. 4, 14).

[16] A. Makhsoos, H. Mousazadeh, S. S. Mohtasebi, M. Abdollahzadeh, H. Jafarbiglu, E. Omrani, Y. Salmani et A. Kiapey, 'Design, simulation and experimental evaluation of energy system for an unmanned surface vehicle', *Energy*, t. 148, p. 362-372, 2018 (cf. p. 4).

[17] M. Lotfi, A. Ashraf, M. Zahran, G. Samih, M. Javadi, G. J. Osório et J. P. Catalão, 'A dijkstra-inspired algorithm for optimized real-time tasking with minimal energy consumption', in *2020 IEEE International Conference on Environment and Electrical Engineering and 2020 IEEE Industrial and Commercial Power Systems Europe (EEEIC/I&CPS Europe)*, IEEE, 2020, p. 1-6 (cf. p. 5).

[18] B. Zhou et Q. Fei, 'Hybrid self-adaptive biobjective optimization of multiple robot scheduling problem for mixed-model assembly lines considering energy savings', *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, p. 0 959 651 820 965 443, 2020 (cf. p. 5).

[19] H. Niu, Z. Ji, A. Savvaris et A. Tsourdos, 'Energy efficient path planning for unmanned surface vehicle in spatially-temporally variant environment', *Ocean Engineering*, t. 196, p. 106 766, 2020 (cf. p. 5).

[20] C. Zhou, S. Gu, Y. Wen, Z. Du, C. Xiao, L. Huang et M. Zhu, 'The review unmanned surface vehicle path planning: Based on multi-modality constraint', *Ocean Engineering*, t. 200, p. 107 043, 2020 (cf. p. 5).

[21] T. I. Fossen, *Handbook of marine craft hydrodynamics and motion control*. John Wiley & Sons, 2011 (cf. p. 6, 13, 31, 41, 43, 44, 48, 50, 51, 54).

[22] G. Canfora, M. Di Penta et L. Cerulo, ‘Achievements and challenges in software reverse engineering’, *Communications of the ACM*, t. 54, n° 4, p. 142-151, 2011 (cf. p. 6, 56).

[23] marcelo paravisi. (). Usv_sim_lsa, adresse : `https://github.com/disaster-robotics-proalertas/usv_sim_lsa`. (accessed: 01.03.2021) (cf. p. 6).

[24] H. C. Brown, L. K. Jenkins, G. A. Meadows et R. A. Shuchman, ‘Bathyboat: An autonomous surface vessel for stand-alone survey and underwater vehicle network supervision’, *Marine Technology Society Journal*, t. 44, n° 4, p. 20-29, 2010 (cf. p. 12).

[25] W. Naeem, T. Xu, R. Sutton et A. Tiano, ‘The design of a navigation, guidance, and control system for an unmanned surface vehicle for environmental monitoring’, *Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment*, t. 222, n° 2, p. 67-79, 2008 (cf. p. 12).

[26] B. Bingham, N. Kraus, B. Howe, L. Freitag, K. Ball, P. Koski et E. Gallimore, ‘Passive and active acoustics using an autonomous wave glider’, *Journal of field robotics*, t. 29, n° 6, p. 911-923, 2012 (cf. p. 12).

[27] J. Jin, J. Zhang et D. Liu, ‘Design and verification of heading and velocity coupled nonlinear controller for unmanned surface vehicle’, *Sensors*, t. 18, n° 10, p. 3427, 2018 (cf. p. 12, 15, 44).

[28] X. Mou et H. Wang, ‘Wide-baseline stereo-based obstacle mapping for unmanned surface vehicles’, *Sensors*, t. 18, n° 4, p. 1085, 2018 (cf. p. 12).

[29] A. J. Sinisterra, M. R. Dhanak et K. Von Ellenrieder, ‘Stereovision-based target tracking system for usv operations’, *Ocean Engineering*, t. 133, p. 197-214, 2017 (cf. p. 12).

[30] Z. Liu, Y. Zhang, X. Yu et C. Yuan, ‘Unmanned surface vehicles: An overview of developments and challenges’, *Annual Reviews in Control*, t. 41, p. 71-93, 2016 (cf. p. 12).

[31] M. A. Abkowitz, ‘Measurement of hydrodynamic characteristics from ship maneuvering trials by system identification’, rapp. tech., 1980 (cf. p. 12).

[32] L. J. Yang et Y. H. Tao, ‘Calculation of the coefficients of ships resistant force in mmg model based on cfd’, in *Advanced Materials Research*, Trans Tech Publ, t. 694, 2013, p. 605-613 (cf. p. 13).

[33] T. Fossen, ‘Guidance and control of ocean vehicles. john willey & sons’, *Inc., New York*, 1994 (cf. p. 13).

[34] M. Wahab, F. Rios-Gutierrez et A. El Shahat, *Energy modeling of differential drive robots.* IEEE, 2015 (cf. p. 13).

[35] L. Hou, L. Zhang et J. Kim, ‘Energy modeling and power measurement for mobile robots’, *Energies*, t. 12, nº 1, p. 27, 2019 (cf. p. 13).

[36] L. Hou et al, ‘Energy modeling and power measurement for three-wheeled omnidirec- tional mobile robots for path planning’, *Electronics*, t. 8, nº 8, p. 843,2019(cf. p. 13).

[37] S. L. Canfield, T. W. Hill et S. G. Zuccaro, ‘Prediction and experimental validation of power consumption of skid-steer mobile robots in manufacturing environments’, *Journal of Intelligent & Robotic Systems*, t. 94, nº 3-4, p. 825-839, 2019 (cf. p. 13).

[38] M. F. Jaramillo-Morales, S. Dogru, L. Marques et J. B. Gomez-Mendoza, ‘Predictive power estimation for a differential drive mobile robot based on motor and robot dynamic models’, in *2019 Third IEEE International Conference on Robotic Computing (IRC)*, IEEE, 2019, p. 301-307 (cf. p. 14).

[39] M. F. Jaramillo-Morales, S. Dogru, J. B. Gomez-Mendoza et L. Marques, ‘Energy estimation for differential drive mobile robots on straight and rotational trajectories’, *International Journal of Advanced Robotic Systems*, t. 17, nº 2, p. 1 729 881 420 909 654, 2020 (cf. p. 14).

[40] C. Li, J. Jiang, F. Duan, W. Liu, X. Wang, L. Bu, Z. Sun et G. Yang, ‘Modeling and experimental testing of an unmanned surface vehicle with rudderless double thrusters’, *Sensors*, t. 19, nº 9, p. 2051, 2019 (cf. p. 15, 45, 46, 56, 57).

[41] H. Niu, Y. Lu, A. Savvaris et A. Tsourdos, ‘Efficient path planning algorithms for unmanned surface vehicle’, *IFAC-PapersOnLine*, t. 49, nº 23, p. 121-126, 2016 (cf. p. 15).

[42] D. Mu, G. Wang, Y. Fan, X. Sun et B. Qiu, ‘Modeling and identification for vector propulsion of an unmanned surface vehicle: Three degrees of freedom model and response model’, *Sensors*, t. 18, nº 6, p. 1889, 2018 (cf. p. 15, 44-46).

[43] C. R. Sonnenburg et C. A. Woolsey, ‘Modeling, identification, and control of an unmanned surface vehicle’, *Journal of Field Robotics*, t. 30, nº 3, p. 371-398, 2013 (cf. p. 15, 46).

[44] S. Wirtensohn, J. Reuter, M. Blaich, M. Schuster et O. Hamburger, ‘Modelling and identification of a twin hull-based autonomous surface craft’,

in *2013 18th International Conference on Methods & Models in Automation & Robotics (MMAR)*, IEEE, 2013, p. 121-126 (cf. p. 15).

[45] J. Kramer et M. Scheutz, ‘Development environments for autonomous mobile robots: A survey’, *Autonomous Robots*, t. 22, nº 2, p. 101-132, 2007 (cf. p. 16).

[46] L. Žlajpah, ‘Simulation in robotics’, *Mathematics and Computers in Simulation*, t. 79, nº 4, p. 879-897, 2008 (cf. p. 16).

[47] L. Pitonakova, M. Giuliani, A. Pipe et A. Winfield, ‘Feature and performance comparison of the v-rep, gazebo and argos robot simulators’, in *Annual Conference Towards Autonomous Robotic Systems*, Springer, 2018, p. 357-368 (cf. p. 16).

[48] N. Koenig et A. Howard, ‘Design and use paradigms for gazebo, an open-source multi-robot simulator’, in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, IEEE, t. 3, 2004, p. 2149-2154 (cf. p. 16).

[49] Z. B. Rivera, M. C. De Simone et D. Guida, ‘Unmanned ground vehicle modelling in gazebo/ros-based environments’, *Machines*, t. 7, nº 2, p. 42, 2019 (cf. p. 17, 84).

[50] M. M. M. Manhães, S. A. Scherer, M. Voss, L. R. Douat et T. Rauschenbach, ‘Uuv simulator: A gazebo-based package for underwater intervention and multi-robot simulation’, in *OCEANS 2016 MTS/IEEE Monterey*, IEEE, 2016, p. 1-8 (cf. p. 17, 84).

[51] M. Zhang, H. Qin, M. Lan, J. Lin, S. Wang, K. Liu, F. Lin et B. M. Chen, ‘A high fidelity simulator for a quadrotor uav using ros and gazebo’, in *IECON 2015-41st Annual Conference of the IEEE Industrial Electronics Society*, IEEE, 2015, p. 002 846-002 851 (cf. p. 17, 84).

[52] C. Fairchild et T. L. Harman, *ROS Robotics By Example: Learning to control wheeled, limbed, and flying robots using ROS Kinetic Kame*. Packt Publishing Ltd, 2017 (cf. p. 19, 67, 69).

[53] P. Corke, ‘Integrating ros and matlab [ros topics]’, *IEEE Robotics & Automation Magazine*, t. 22, nº 2, p. 18-20, 2015 (cf. p. 24).

[54] W. Touzout, Y. Benmoussa, D. Benazzouz, E. Moreac et J.-P. Diguet, ‘Unmanned surface vehicle energy consumption modelling under various realistic disturbances integrated into simulation environment’, *Ocean Engineering*, t. 222, p. 108 560, 2021 (cf. p. 84).