Universität Rostock

Traditio et Innovatio

LIÈGE université

CENTRALE NANTES

POLITÉCNICA

icam

SOLENT UNIVERSITY SOUTHAMPTON

UNIVERSITAS GALATIENSIS

UNIVERSITÀ DEGLI STUDI DI GENOVA

West Pomeranian University of Technology in Szczecin

With the support of the Erasmus+ Programme of the European Union

KER YACHT DESIGN

Preliminary sail shape optimisation using neural network aerodynamic and hydrodynamic models

submitted on :	04 August 2020
Handed in by:	Pierre Courtel
	5 cité Oricher Hoel, L-8036 Strassen, Luxembourg
	courtel.pierre@gmail.com
Enrolment number:	219 200 146
Supervisor:	Prof. Jean-Baptiste R. G. Souppez
Primary Reviewer:	Prof. Dr.-Ing. Sascha Kosleck
	Chair of Ocean Engineering at Rostock University
	Oskar-Kellner-Institut (OKI) / Raum 104, Justus-von-Liebig-Weg 2, 18059 Rostock, Germany
Secondary Reviewer:	Prof. Florin Pacuraru
	Dunarea de Jos University of Galați
	Department of Naval Architecture, 47 Domnească Street, 800008, Galați, România

EMship+ Advanced Design

Master Thesis

Printed on acid-free and elemental chlorine-free bleached paper.

As of:: October 1, 2020

Abstract

Quantifying the performance of a sailing vessel can be a challenging task. Velocity Prediction Programs are numerical tools created to model the physics of yachts. Their behaviour can be decoupled in two major components, the aero- and hydrodynamic forces and moments. The main algorithm employed is in charge of balancing the forces and moments of the two components for each degree of freedom. Once an equilibrium is found, the forces can be used to determine a large range of outputs, the speed of the boat being the most important one.

The main goal for this master thesis is to develop a website to manage a cloud-based VPP. A functioning prototype has been implemented allowing multiple users to access the VPP online. The VPP has been tested against empirical models and forces have been shown to respect an order of magnitude although not matching precisely. A use-case scenario for the platform was given with the study of various keel designs leading the speed increases of around 5%.

This application opens high-end technologies to a new market at an affordable cost, without the need to invest in computers. It creates opportunities for design offices, such as Ker Design where the internship was conducted, to turn in-house developed software into product for a new range of customers.

Acknowledgements

First and foremost, I would like to thank everyone at Ker Yacht Design for welcoming me into their team during these five months despite the pandemic. A special thanks to Jason Ker for his guidelines and expertise. I also want to thank Marcus Mauleverer for his invaluable help and support while setting up the VPP and website.

Then, I would like to thank Jean-Baptiste Souppez who helped me to find this internship and provided continuous support throughout the duration of this project. His patience and his time have been very appreciated. His advice has always been valuable to me.

I would like to thank my reviewers Prof. Dr.-Ing. Sascha Kosleck from the University of Rostock and Prof. Florin Pacuraru from the Dunarea de Jos University of Galați.

Finally, I would like to thank my parents and family for their moral and financial support throughout this master program.

Contents

Declaration of Authorship

I, Pierre COURTEL, declare that this thesis and the work presented in it are my own and have been generated by me as the result of my own original research.

Where I have consulted the published work of others, this is always clearly attributed.

Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.

I have acknowledged all main sources of help.

Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

This thesis contains no material that has been submitted previously, in whole or in part, for the award of any other academic degree or diploma.

I cede copyright of the thesis in favour of the University of Rostock.

Luxembourg, the October 1, 2020

Signature:

List of Figures

List of Tables

List of code Listings

Nomenclature

Variable	Full Name
AR	Aspect Ratio
AWA	Apparent Wind Angle
AWS	Apparent Wind Speed
BAS	Boom Above Sheer
BOA	Beam overall
BWL	Beam Waterline
Cd	Drag coefficient
Cl	Lift coefficient
E	Mainsail Foot
EHM	Effective Mast height
FA	Average Freeboard
F_D	Drive Force
HLP	Headsail luff perpendicular
I	Height of Foretriangle
J	Base of Foretriangle
P	Mainsail Hoist
SSF	Sail Side Force
T	Draught (Keel included)
T_C	Canoe Draught (Keel NOT included)
TWA	True Wind Angle
TWS	True Wind Speed
VMG	Velocity Made Good

Variable	Full Name
ϕ_{mast}	Average Mast diameter
x	Surge direction
y	Sway direction
z	Heave direction
β	Yaw angle
δ	Rudder angle, Leeway
θ	Pitching angle
ρ_{air}	air density
φ	Rolling angle

Accronym	Full Name
aws	amazon web services
CFD	Computational Fluid Dynamics
CSS	Cascading Style Sheets
DOF	Degrees of Freedom
DSYHS	Delft Systematic Yacht Hull Series
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
I.T.	Information technology
IDE	Integrated Development Environment
ITTC	International Towing Tank Conference
ORC	Offshore Racing Congress
OWASP	Open Web Application Security Project®
PBKDF2	Password-Based Key Derivation Function 2
RANS	Reynolds-averaged Navier–Stokes equations
SQL	Structured Query Language
URL	Uniform Resource Locator
VPP	Velocity Prediction Program

Chapter 1

INTRODUCTION

Predicting the speed of sailing vessels is an essential technique to have for naval architects. When inquiring for new yachts or during a refit, customers have demands regarding the performance of their boats. For high-end skippers, the wish is to perform on race day and ultimately win regattas while other customers might have comfort while under sail as a priority. In both cases modelling the physics of the system is required to understand how the yacht behaves in its environment.

Velocity Prediction Programs attempt to create models which represent the reality of sailing through thorough application of cutting edge engineering and science. Common VPPs are built on two core components. Firstly, the aerodynamic module in charge of determining the driving force and heeling moment of the vessel as a consequence of the air flowing on the sails, hull and rig. Secondly, the hydrodynamic module responsible to compute the resistance forces and righting moment of the vessels due to the hull and appendages passing through water. Additional modules must be considered, to cover the stability of the craft for example, but the aero- and hydrodynamic modules are the ones studied in particular for this thesis. Balancing algorithms are then implemented to establish the equilibrium between the forces and moments of the two models. The speed of the boat can then be calculated thanks to these forces.

The main goal of this thesis is to create a web interface to deploy a VPP on a cloud-based computer service. The motivation driving this project is to create a new service allowing customers to run VPPs from the web. The innovation being that the users are no longer required to invest in state-of-the-art computers as the calculations are outsourced to large servers where economies of scale allow for competitive pricing.

The VPP used for this project has been developed over the years at Ker Design, the models are based on CFD data and feature state-of-the-art neural network interpolation schemes to allow time-efficient model taking up to 15 parameters as inputs. The VPP in its current state is based on the CFD results for a well researched test vessel of length 40ft, a 3d view of the hull geometry is depicted on 1.1. The website is built on the python micro framework called flask to allow convenient communication between the website and the python

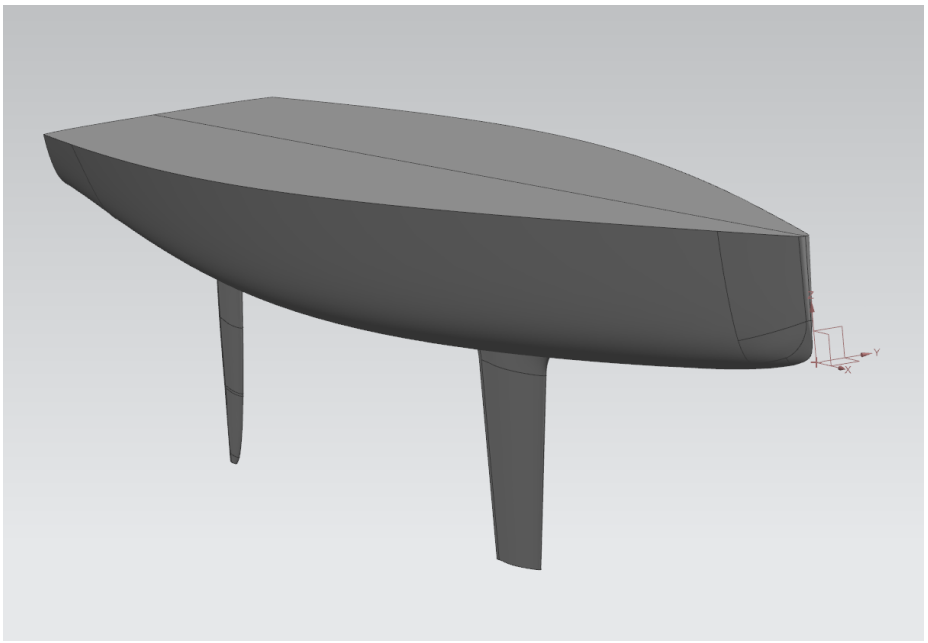

Figure 1.1: 3d picture of the 40ft hull used for the CFD hydrodynamic calculations.

scripts forming the VPP. Security concerns have been integrated to the project and appropriate measures have been taken to prevent data from being leaked on the web.

The chapter 2 is dedicated to the literature review, the presentation of the website is given in chapter 3. In the chapter 4 the actual VPP currently deployed on the website and the models implemented are detailed, a performance analysis of the python scripts has also been given. A use case describing how the platform can be used for design optimisation is given in chapter 5. Finally, the conclusion is found on 6.

Chapter 2

LITERATURE REVIEW

In this chapter

2.1 Velocity Prediction Programs

A sailing boat consists of a very complex system operating at the interface of two fluids: water and air. A simple example to illustrate the performance prediction problem is described by Fossati [1]. Say the designer wants to reduce friction resistance of a hull by decreasing the wetted surface area while keeping the waterline length unchanged. The beam has to be reduced which consequently lowers the transverse stability, thus for a given sail area the heel increases. This new heel changes the geometry of the immersed hull volume affecting the resistance and modifying the forces produced by the sails.

With this simple example one can get a sense for the difficulty of the problem and why numerical tools are key to solving it. The role of the VPP is to assess the performance gains associated with changes in the key design parameters of a boat (length, beam, displacement…) as well as finer settings while on the water (sail plan dimensions, reefs, ballast, crew position…). One of the first VPPs to be successfully implemented came out in the '90s with the work of Van Oossanen [2].

The continuous progress seen in the field of computer science have allowed for more and more complex problems to be solved within a feasible time frame. VPPs have also benefited from these advances, solving the equilibrium equations for all degrees of freedom (DOF) was not something possible 30 years ago [3] whereas now, computers are powerful enough to do so.

2.2 History of the VPP

Performance prediction is an essential topic for boat designers. Its origin dates to the beginning of the 20th century. According to Böhm [4] the rise of yacht racing as a sport at that time, associated to the complexity of sailing vessels, has driven researchers to better understand the physics behind sailing. The first hypothesis to consider is to divide the problem into various models, including hydrostatics and stability, but with the two most prominent being, one for the hydrodynamic forces and a second one for the aerodynamic forces. The performance can be evaluated when all DOFs considered are in equilibrium.

2.2.1 Hydrodynamic Investigations

William Froude is the precursor for the description of the mechanisms involved around the waves generated by hulls moving through water. His work allowed to conduct model sized tests in tanks that are comparable to the full-scale behaviour. Originally intended for merchant vessels and navy ships, George L.Watson documents the first experiments to improve the performance of America's cup yachts in 1901 .

In 1936 K. Davidson published the first successful towing tanks for yachts. Considering the forces of the sails he determines approximately the hull resistance [5]. Later in 1974 the Delft Systematic Yacht Hull Series (DSYHS) was created at the Delft University of Technology. The ever ongoing study investigates the influence of hull form parameters on the resistance. Towing tanks tests were originally performed in the '70s for a few dozens of hull shapes with only certain parameters changing from one to another [6]. The results yield to a regression model that can be used to compute the hull resistance and its appendages thanks to their parameters (parameters within a precise range). The research on the series is well documented and continues to be regularly improved. New hulls have been added over the years to the series and the regression formulas have regularly updated leading to the latest publication by Keuning [7]. This method is convenient during the design phase as the hull form is not required.

For hulls outside the provided range, or for more complex appendages, the resistance can't be approximated properly with DSYHS and other numerical tools have to be employed. Amongst the favourite in use today, the *Computational Fluid Dynamics* (CFD) was first introduced the world of yachting for hydro purposes at the 1983 America's Cup as Böhm [4] suggests. This historic campaign saw the Challenger, Australia II, take the victory from the New York Yacht Club undefeated since the creation of the cup 132 years prior. The innovative keel design of the challenger resulting of CFD computations launched a controversy and initiated a trend towards numerical simulations.

Recent technologies for hydrodynamics are discussed in depth in sections 2.4 outside of this historical overview.

2.2.2 Aerodynamic Investigations

Assessing the sail forces has been quite a challenge for a long time. One of the first attempts has been mentioned already with the work of Davidson in 1936 [5]. During his towing tests experiments Davidson uses full-scale measurements to apply forces to his model sized hulls. He also establishes the Gimcrack set of coefficients used to characterise rig forces. In the discussions of the paper, the accuracy of the coefficients was in question. Although rudimentary compared to today's standards, the gimcrack has been applied at the time to several J-boats with no major difficulties uncovered. They appear to be successful and a good first approximation. Another interesting comment was raised in the discussion by Edward P. Warner an aeronautical engineer at the Massachusetts Institute of Technology. Having studied yacht sails in the wind tunnel [8], Warner was handicapped by the lack of knowledge on the hydrodynamics of the hull. In his view Davidson is also handicapped by a simplistic aerodynamic model and he hopes that future research can link the two laboratories together and carry the knowledge far beyond what is known at the time.

Wind tunnel testing continues to be the standard research facility for study of yacht sails. In the '70s C.A. Marchaj's work at the University of Southampton contributes greatly to the knowledge. The addition, in the '90s, of curved veins inside the wind tunnel creating the twisted flow have been a major discovery and massively improved the correlation from the model world to the full-scale sails. In 1995 the University of Auckland built the first Twisted-Flow wind tunnel to investigate the 29th America's cup challenger Team New Zealand [9].

In parallel to model sized tests, full-scale measurements have also been undertaken on the water. In the '80s researchers at the Massachusetts Institute of Technology built dynamometers to measure the rig forces [10, 11]. The boat was 35 ft scale-down version of

83 ft Maxi Sloop which had already been tested in a towing tank. The data fed a VPP with results coherent to on water performance.

In 1968 Milgram [12] developed one of the first numerical models for yacht sails and with increasing computational power made available in the late 20th century, CFD has also been used to compute aerodynamic forces. Viola and Flay [13] presented a comparative study of pressure distributions on sails with three methods, wind tunnel testing, full-scale measurements and numerical simulations. A good agreement was reported at the time between the techniques. The authors are still conducting research to improve the modelling of sails and have later found the wind tunnel experiments were actually wrong [14]. Nowadays computers can handle the full 3d model of a yacht allowing for optimisation of the boat in the digital world. The second approach is based on an implicit model; here the optimal sail trim has been found and penalty system is put in place to account for the changes in the sail shape.

Latest advances in aerodynamic are being discussed further in section 2.5.

2.3 Degrees of Freedom

Yachts are exposed to 6 degrees of freedom (DOF), 3 translations (surge x, sway y and heave z) and 3 rotations (roll φ, pitch θ and yaw β). The degrees of freedom are depicted by fig. 2.1.

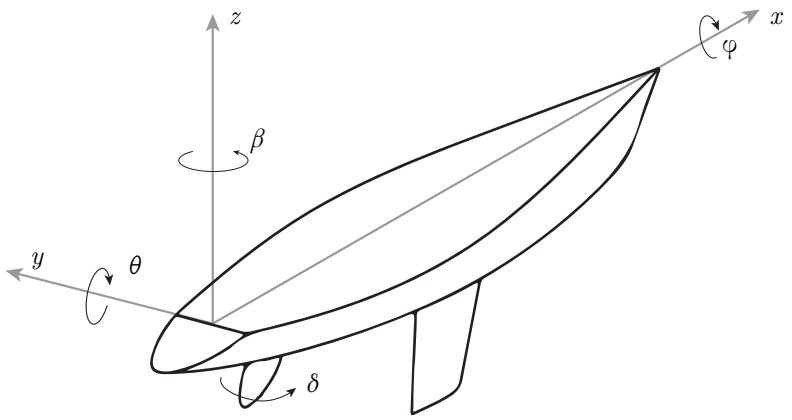

Figure 2.1: Coordinate system and the 6 degrees of freedom of a yacht, edited from [15]

2.3.1 Presentation of the DOFs

2.3.1.1 Translations

Surge (x) Surge characterises the longitudinal motion of the boats or, in other words, the speed of the boat. This is the most important direction to consider in VPP applications.

Sway (y) Sway corresponds to the transverse movement of the yacht. It can be seen as the drift of the vessel from its course. It is also known as leeway and is a key component. Indeed the leeway angle defines the angle of attack for the keel which has to generate the lift to counteract the sail side force.

Heave (z) Heave is responsible for vertical motion of the boats. With traditional Archimedean i.e. subject to hydrostatic lift, the motion is balanced naturally between the weight of the boat and the buoyancy force. The impact of the heave is often minimal on the performance and therefore it can be neglected. For modern crafts with dynamic lift introduced at high speeds when the hull is planning or when hydrofoils are employed, the heave motion is not negligible and must be taken into consideration.

For most mono hulls, the motion is a balance between the weight of the boat and the buoyancy force. The impact of the heave is often minimal on the performance and therefore it can be neglected.

2.3.1.2 Rotations

Roll (φ) The roll or heel is also essential to the performance of a sailing yacht, the rotation about the longitudinal axis has impacts on the stability, the aerodynamic forces and on the hull resistance.

Pitch (θ) The rotation about the transverse axis is minimal for long monohulls. The longitudinal stability being large small rotations are observed with minimal impact on the performance.

Yaw (β) This rotation is of relative importance, the moment resulting from the yaw rotation is counteracted by the rudders. This action comes at the cost of a small drag penalty that affects the overall performance.

2.3.2 Forces and Moments Acting on Yachts

The forces and moments acting on the system are always in equilibrium. The actions along the main 6 degrees of freedom are given in figs. 2.2 & 2.3.

Along the surge direction, the Driving Force F_M from the wind acting on the rig is in opposition with the resistance of the water acting on the hull. Leading the following equation:

$$F_x = F_M - R_{tot} = 0 \tag{2.1}$$

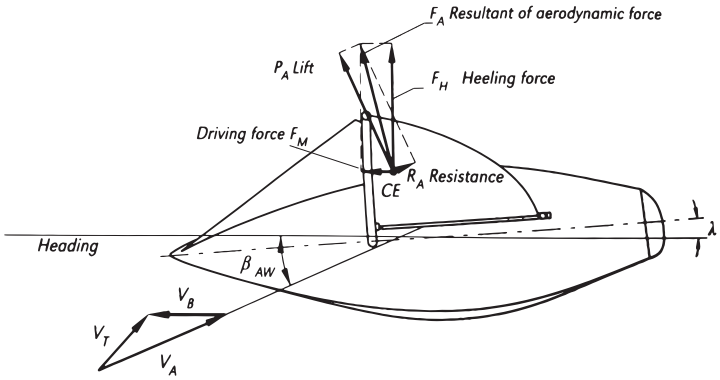

Figure 2.2: Definition of the aerodynamic forces, edited from [1]

As for the sway the equilibrium is a balance between the side forces of the wind on the rig and of the water on the keel, the equation writes:

$$F_y = F_{lat} - P_{lat} = 0 \tag{2.2}$$

The rolling moment equilibrium sees the heeling moment created by the wind opposed by the righting moment created by the mass the keel rolling. It comes:

$$M = M_{heeling} - M_{righting} \tag{2.3}$$

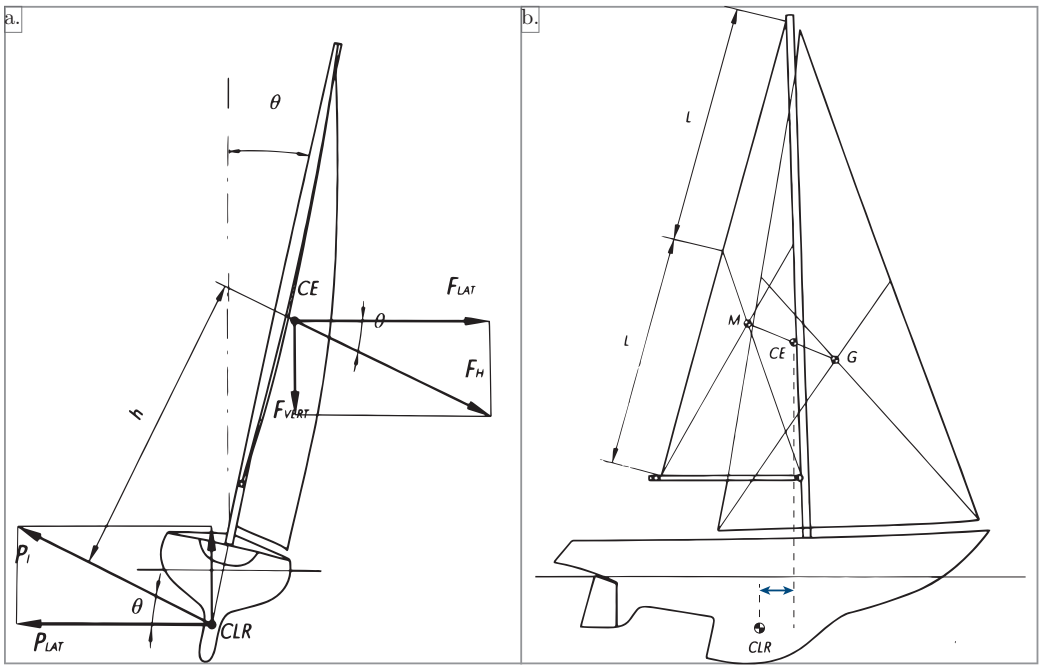

Figure 2.3: a. Definition of the Lateral forces while the yacht is heeling, b. Position of the centre of effort and the centre of lateral resistance, edited from [1]

These three fundamental equilibria are often coupled to an additional DOF, yaw. A moment about the z-axis can occur if the centre of effort of the driving force and the centre of effort of the resistance are not aligned on the longitudinal axis. Fig 2.2 b. shows two centres that are slightly off. The moment created must be counterbalanced by an action at the rudder at the cost of a small drag penalty.

The two remaining degrees are mostly relevant for foiling crafts but can also be taken into account for monohulls. Guell and Souppez have used existing 3 DOF VPPs to incorporate the use of foils [16] while new VPPs have been created to enhance capabilities. [14]

2.3.3 Accuracy of the VPP

As with any method the output is only as representative as the input is. The VPP algorithms rely on qualitative models the more accurate they are the closer the performance will be to on-water measurements. [17]

2.4 Hydrodynamic Models

Understanding the behaviour of vessels in the sea has been a topic of research for a long time. The first models are attributed to Froude [18] and Reynolds [19] their groundbreak-

ing discoveries form the fundamental knowledge that eventually lead to the International Towing Tank Conference (ITTC) 1957 friction coefficient [20].

Still in use today the ITTC coefficient is particularly applied in the most important systematic series for yachts, the DSYHS, that started with Gerritsma in the '70s and continues to this day [7]. Various hull forms have been tested in towing tank facilities at the Technical University of Delft. The hydrodynamic results have then been compiled and a regression scheme was generated to model all sorts of hulls within strict range of parameters. The strength of this empirical method lies in its simple implementation, however, there are limitations as Souppez pointed out [17].

Other hydrodynamic models exist. The Offshore Racing Congress (ORC) is in charge of establishing a handicap system for yachts. The purpose is to rate a fleet of boats from different sizes in such a way that everyone at the start has the same chances of winning. To achieve their goal the ORC has built a VPP to establish the performance of the sailing boat [21]. The hydrodynamic model in this method is based on a regression analysis of towing tank experiments, similar to the systematic series studied in Delft.

2.5 Aerodynamic Models

When it comes to aerodynamics, the research was first primarily focused at the aeronautic industry and the study of aeroplane wings. Naval architects have therefore tried to use this knowledge and to apply it to sailing yachts. Sails can, however, behave in ways completely different from aeroplane wings. While sailing downwind headsails like the spinnaker possess a large curvature which disrupts the flow creating separation.

Downwind modelling of the spinnaker has been one of the research topics of Professor Viola at the University of Edinburgh, leading to several papers on the subject [22, 23]. During this research period a new behaviour around downwind sails, leading edge vortices, have been observed. This phenomenon was later confirmed on model scale tests in the wind tunnels [24, 25]. Souppez, Arredondo-Galeana & Viola have further investigated the numerical representation of the flow separation on downwind sails [14].

When it comes to VPP, there are two major approaches to the aerodynamic model according to Böhm [4]. The first is composed of an explicit sail shape where the VPP has access to the entire set of sails available for the boat and their corresponding force coefficients. Proceeding like this allows for the VPP to find the optimal sail shape for a given wind and boat condition. While the model gives a very accurate representation of sails at sea with individual coefficients for each configuration, the method requires a substantial amount of

CFD calculations and is usually reserved for highly competitive sailing teams. The second approach uses an implicit model, with the optimal sail already chosen a set of penalties can be applied to differentiate the sail shapes. In this was the sail shape and aerodynamic model are separated, this simplification relates to less computational power required with sufficient accuracy for preliminary design. This second technique is based on the Kerwin model which has been significantly improved by Hazen [26].

Derived from the previously mentioned coefficients, the ORC VPP aerodynamic model is well-known method in the yachting industry. The output of the VPP is sometimes contested as some boats see clearly on water benefits when compared with others at equivalent numerical predictions. Over the years several updates have been implemented to the ORC VPP. For the upwind scenario Mauleverer [27] has found the ORC model to be inaccurate and provided adjustments to the system to improve it. Discrepancies were also found between the ORC model and experimental studies by Focarile and Souppez [28]. Finally, for the downwind scenario, recent work has shown that established wind tunnel practises are actually incorrect, thus shedding doubt on the accuracy of coefficients [14]. All things considered the coefficients are questioned yet still considered an established practice. The convenience of use in VPP application prevails the doubts that have been raised.

2.6 VPP for Design Optimisation

For most naval architecture project, the boat speed is a contractual agreement between the customer and the designer. As such getting it right is a critical requirement. Throughout the design phase VPPs can be used precisely for that purpose. At the preliminary stage empirical methods such as DSYHS and ORC Aero can be employed, they offer relatively simple equations while not requiring the full geometry. Parametric investigations can be tested against each other with ease. More advanced models are necessary at later stages to validate the empirical results.

VPP can be applied to compare two generations of boats in the same class with noticeable gains in performance [29]. Because the two generations have similarities between them once the first VPP is set up properly, the additional cost associated to the implementations of the second iteration of boat is relatively low. This is not always the case when design options are to be compared and several models of the boat need to be built as Thomas and Souppez have shown [30]. In light of these limitations, there is a high interest in improving performance at early design stage by using more advanced numerical methods to speed up the process.

2.7 Surrogate Models

A VPP relies on input data for the aero- and hydrodynamic forces, the former have to cover a range of apparent wind angels (AWAs) and the latter range of boat speeds. The data points are then interpolated with curves or surface fitted schemes to find the forces used in the VPP. The accuracy of the VPP heavily depends on the input data. These points have traditionally been generated with extensive CFD campaigns at a cost that is only affordable to established sailing team, research laboratories and high-budget projects not for the sail-making industry.

That is where surrogate models come in, particularly based on artificial intelligence and neural networks in this project. It's been shown that complex functions that normally require hours or even days of simulation processing time can be reduced to a function evaluated in seconds [31]. The surrogate model can predict the value of a function at points that are unknown. With this technique new data points can be interpolated from a reduced set of numerical data making it ideal for VPP implementation [32].

In his dissertation Mauleverer presented an aerodynamic model with 14 input parameters to investigate [27]. Running each combination of the 14 parameters through CFD would cost a tremendous amount of time and money while not necessarily adding any accuracy to the results. Instead a set, limited in size, of quasi-random points in the 14-dimension space is generated with a sobol sequence. The points are then used to train a Neural Network that will be used afterwards to interpolate new points in the 14-dimension space.

2.8 The Research Question

The design brief for this project was to develop an online tool to give a sailmaker access to a VPP for preliminary design of sail shapes. The team at Ker Design have already programmed or at least partially programmed the individual components for this project. The VPP built around surrogate neural network models for the aero- and hydrodynamic forces has been implemented, and a general presentation website was also completed in-house. Combining the two has never been done in the company so the main goal of this project is to go around the loop at least once and have a functioning prototype of the platform. The development of the website is presented in the first part of this report, the execution of the VPP is discussed in the second part while the third part will present a Use Case of the platform as it stands today.

Chapter 3

PART 1: BUILDING THE WEBSITE

In this chapter

A key innovation of this project lies in the cloud-based interface. The idea is to allow customers to control the VPP from any computer with an internet connection and a browser. The cloud became popular to the consumer market around 2010 with online storage solutions such as Dropbox, Google's Drive, Microsoft's OneDrive, Apple's Icloud and many others. In 2020 the clouds main market targets businesses, with on-demand solutions of computer system resources. Amazon was the first company to offer web services back in 2006, and today many design offices are using these platforms to outsource their complex CFD simulations runs. This chapter presents the website in depth with a description of the functionalities available to the users.

3.1 Python Website Framework

The website created for this application is built on the flask micro-framework. Having built their website on this platform, the engineers at Ker Design (company hosting the

internship) have acquired knowledge on the flask system which gave the author a solid base for the project. Furthermore the Neural Network surrogates models built at Ker Design for their VPP needs are written in python. Flask being written in python it facilitates the links from the website to the python VPP core.

Web development is a fairly new topic in the python language, traditionally building a website involved hiring an I.T. specialist to code the application or using online services with prebuilt pages to fill manually. With this micro-framework, python fluent engineers can create a website in its entirety, maintain the control and update in the future with reasonably small amount of learning.

3.2 Fundamentals of Web Development

A website can be seen as an interface between two computers the client and the server. The client or user is referred to as the front end while the server corresponds to the back end. The front end mainly deals with the *Hypertext Markup Language* (HTML) language in the browser to display information to users. While, the back end is written in python. The client side can send data to the server with the *Hypertext Transfer Protocol* (HTTP) method called POST . This method will mainly be used to submit forms. Upon such actions the server can execute tasks and then reload the HTML page. The essential action for the project is to receive a POST request from the client and then launch a VPP instance on the server. After that post processing actions can also be added to view the results.

3.2.1 Synchronous and Asynchronous Request

A first synchronous request was already presented in the previous paragraph with the POST method. It is characterised as such because the HTML page needs to be completely reloaded after the action. In practice this means that the user sees a freeze on the screen while waiting for the action to complete and the page to reload. On the other hand, an asynchronous request can be used to run background actions without affecting reloading the full page and updating only what needs to be changed. JavaScript is the programming language used in the browser to perform these asynchronous actions.

3.2.2 Bootstrap Templates

Alongside HTML and JavaScript, a third component forms the core of all web applications, the CSS for Cascading Style Sheets. This language has its name suggests, takes care of the

style (e.g. fonts, colours, spacing) and thus the graphical design of the web page. With the rise of open source templates, programmers tend to stay away as much as possible of the CSS. In this context the bootstrap front end open source toolkit was used to build responsive and interactive pages [33]. This toolkit comes with dedicated JavaScript and CSS libraries and has prebuilt elements such as buttons, drop-down menus, tabs menus, tables, and many more options.

3.2.3 HTTP Cookie

A cookie contains data sent from the server to be stored on the browser of the client computer. They are designed primarily to as a reliable mechanism for websites to remember stateful information. For this application the cookie will mainly be used to save the name of the user and whether or not the user is logged in to the service. Additional data is stored on the cookie to populate default values in forms when loading the input tab, for example.

One last very important topic needs to be introduced before the description of the application can begin, security.

3.3 Website Security

Intending to allow multiple users to access the application to run and manage VPP calculations, the website needs a user management system. These days most websites have accounts with functions to sign up, log in, log out, and access zones separated with subscription schemes of various sorts. Although these systems are in place everywhere, no systematic library exists to implement the user accounts. It is the responsibility of the programmer to create a secure environment.

The Open Web Application Security Project® (OWASP) [34] provides excellent resources to protect both users and website developers. The author followed their recommendations to the best of his abilities.

3.3.1 Database Storage

The account information is stored in a table contained within a database managed with the open source MySQL system. The information stored for each user is given on Table 3.1.

Individual users can be identified by three means, looking for their username, email or userId. This userId is an internal reference that is not known to the user. The password will be discussed in the next subsection. The settings can be used to save personal preferences for now it is left empty but it's nice to have the option for future developments. Finally, the rank is there to differentiate the status of the users, customer, technician, administrator, programmer. These status restrict the access to parts of the website and will be described later in the report.

Table 3.1: Description of columns in the SQL users table.

Columns	Type
User ID	Integer
Username	Characters
Password	Hashed Characters
Email	Characters
Settings	Characters
Rank	Integer

Hackers are known to attack SQL databases of websites with what is known as SQL injection. If a form is used to write data to the SQL database, the entry field can be a backdoor to input SQL code right after the data. An example would be to type: 'Pierre Courtel; DROP DATABASE kerWebsite;'in the username field of the login page which would delete the kerWebsite database. To prevent these actions forms fields have to be checked and filtered before passing the data to the database.

3.3.2 Password Hashing and Associated Functionalities

Authentication is an essential step to ensure the security of any applications. With a new registration, users input their personal information and password into a form. Upon submission the data is stored onto the database, at this very step the first security measure is taken. On submission the password is directly hashed. Hashing in opposition to an encryption is a one-way function (i.e. it is impossible to 'decrypt' a hash and obtain the original value), whereas encryption is a two-way function [35]. This process only allows to test whether a string corresponds to the hash stored in the database. The algorithm in place to hash the password is PBKDF2 as per recommendation of OWASP.

While being a secure way to store the password this system creates complications when the user forgets the password. Popular on all user management systems, the password

recovery function can be threat if not implemented properly. In the past software administrators were often in charge of renewing credentials manually when user asked for a new password. This action requires that a person other than the user knows the password, it's a danger and it should be avoided.

Changing the password should be done by the user without relying on anyone else to enter the password in the system. To do so temporary access tokens can be created and sent to the account email with a dedicated URL argument. When clicked the URL token is validated in the back end and the new password form is loaded for the user.

Such a system has not yet been added to the platform for several reasons. The email communication needs to be a no-reply account. If not, should the user reply to the reset password email anyone with access to the email (administrators) can reset the password, creating such an email account is not straightforward and requires an advanced mail server. The tokens are also very sensitive to program the author not being specialised in web security it would have taken additional time to research on the topic. Finally, the application itself being in development for only a hand-full of users, a manual system is put into place. The administrator can set a temporary password for the user, this password is used to access a change password page where the user types in the new secure password.

3.4 General Description of the Software

This section goes through the different functions available to the user has the page loads for the first time. The objective here is to show the front-end graphics while the back end actions will be discussed in the following section.

3.4.1 Home Page

The graphics on the home page are for now blank and only show the top menu navigation bar given in fig 3.1. Four clickable options are made available, Home, Dashboard, Log in, Sign up. Two empty sections are displayed to make use the space, Pricing and Some admin tasks.

On the first load, the site cookie is empty and thus the user is logged out. With the logged out status the dashboard section is not accessible, a click on this button automatically redirects to the log in page. If the user is new to the platform, it is possible to sign up to the service.

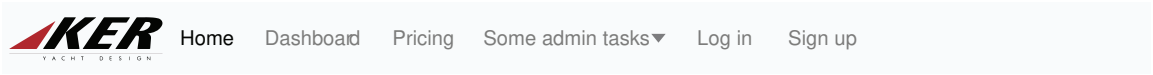

Figure 3.1: Top Menu navigation bar with logged-out status

3.4.2 Sign up

The sign up page is a straight forward form with fields to enter a username, an email and a password as can be seen in fig 3.2. In the background validations have been put in place to prevent insecure accesses.

The username and email need to be unique therefore before saving the new user in the database, a check is performed to make sure no account is linked to the new entries. Should the username be in the database, the sign up page reloads with an error message prompting the user to choose another username. If the email is in the database, the log in page is reloaded with a warning message asking the user to login rather than sign up.

The password validation requires all the following conditions to be verified, if one of the conditions fails an error message notifies the user of the corrections to perform on the password. Typing mistakes can happen, to prevent them the password is asked twice, both should match each other.

- Minimum length 8 characters
- Maximum length 20 characters
- At least one uppercase letter
- At least one lowercase letter
- At least one number
- At least one symbol in the following list: !”# $ % & \’()*+,-./:;¡=¿?@[]^_‘{|}∼

Finally, terms and conditions should apply when signing up to the service so a box needs to be ticked.

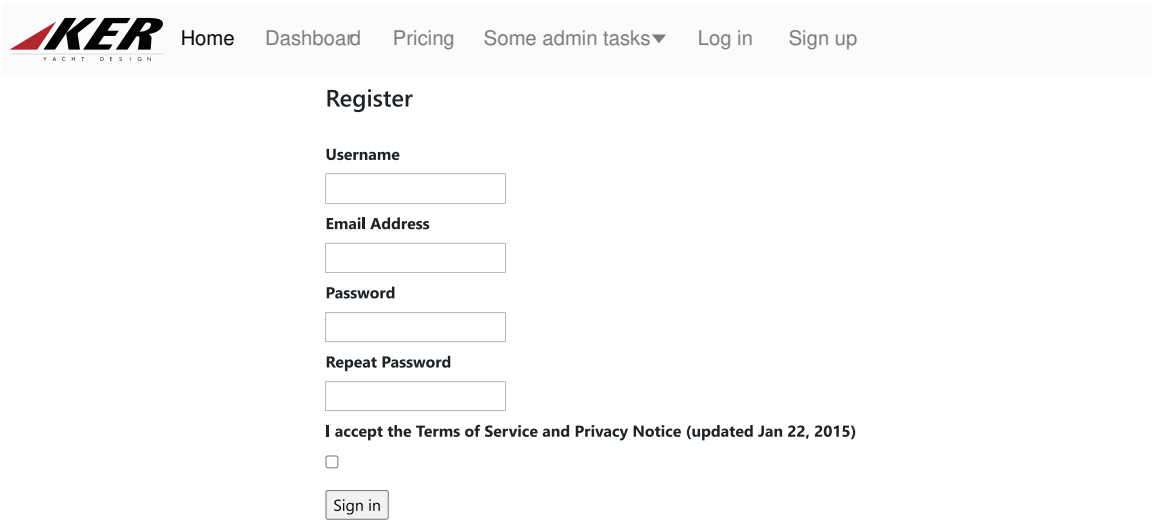

Figure 3.2: Sign up form

3.4.3 Log in

The login page is also straightforward when the log in button is clicked the inputs are checked in the database.

If the username is not in the database or if the password entered does not match the hash associated to the username, the login page is reloaded with an error message: *Invalid credentials, try again*. A mistake often seen on the websites is when the message specifies which field is incorrect giving information to malicious users. Here it is not the case. To prevent brute force attacks a captcha can be asked if the login fails many times in a row.

If the username exists and the password verifies the associated hash saved in the database, the state, logged in, and the username are added to the website cookie. The page is then reloaded to the Dashboard.

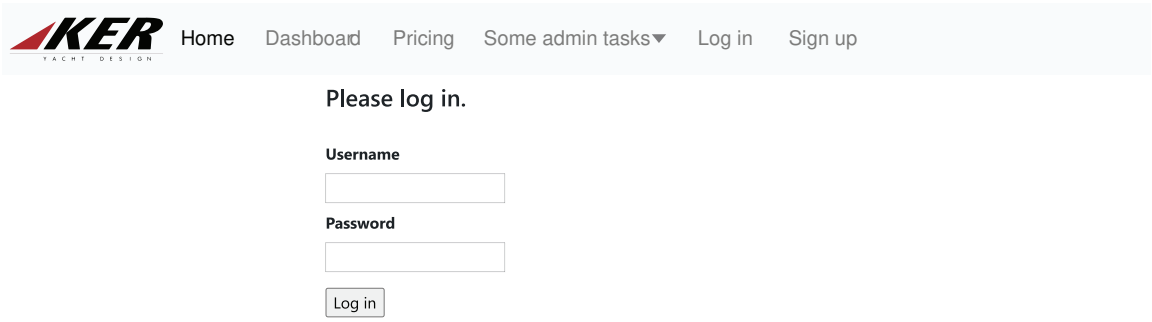

Figure 3.3: Log in Form

Once the state is switched to logged in, the top menu navigation bar is updated to remove the login and sign up buttons. The username appears and new functions are made available in a drop-down menu, Log out, Change Password and Request Password Reset as seen in fig 3.4.

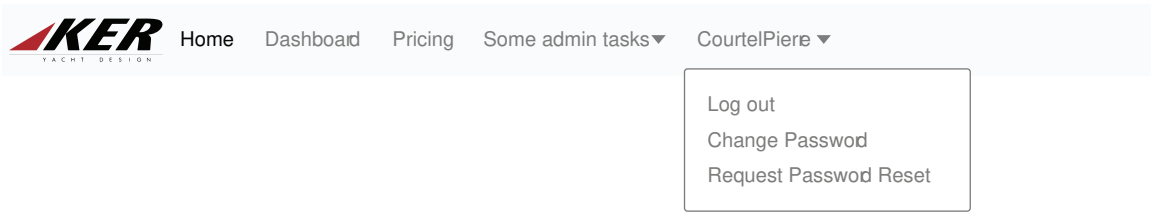

Figure 3.4: Updated nav-bar menu after user CourtelPierre logged in to the platform.

3.4.4 Log out

When the log out button is clicked, the cookie is cleared, automatically switching the status to logged out and the home page is loaded with a warning message: *You have been logged out successfully*

3.4.5 Change Password

To change the password the user is already logged in; therefore the system can retrieve the username from the cookie. The form uses the same validations as the one mentioned in 3.4.2. The old password is asked as an added security. If it matches to the hash in the database and if the new passwords validations are accepted, the database is updated with the new password and the user is logged out.

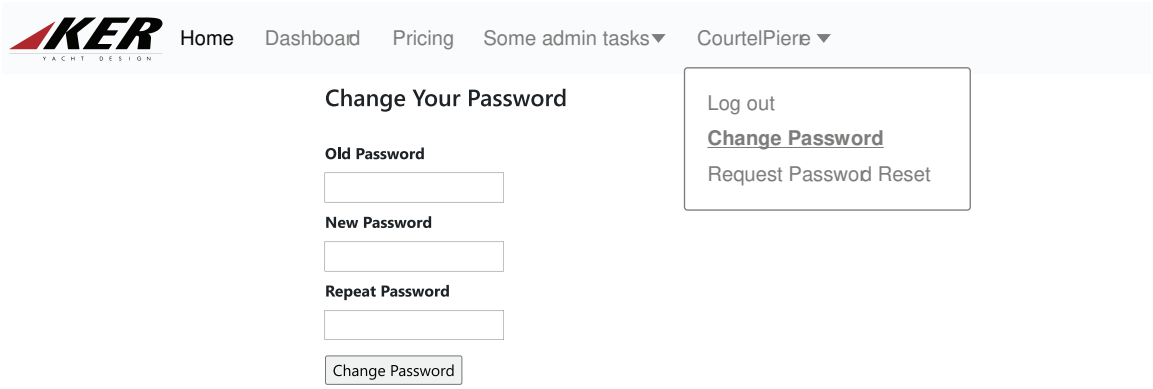

Figure 3.5: Form allowing users to change their passwords.

3.4.6 Request Password Reset

As previously mentioned in the Website Security section 3.3, the password request is still in a temporary state and requires the administrator to manually create a new password in the database for the user.

3.4.7 Dashboard

The main application is located under the dashboard section which is accessible to logged in users only. It is divided in three tabs. The Task Manager to monitor and launch the VPP runs. The inputs tab where is possible to type in data and create new batches and finally the visualisation tab where charts outputs are displayed.

3.4.7.1 Task Manager

The task manager is composed of three distinguished sections separating Pending, Active and Completed tasks. It is displayed in fig 3.6.

The pending tasks are managed with a drop-down selection, the selected task can be started with a click on the run button. Once launched the task becomes active and appears in the active table below. For active tasks, an automatic procedure checks the advancement of the batch and logs it in a progress bar. When finished the status switches to completed and the task is moved to the bottom table.

It is possible to plot charts for a given run in both active and completed status. More on that functionality later in the VPP description.

3.4.7.2 Input Tab

This part of the website gathers the input data and create new batches, fig 3.7 gives the representation for a standard run. When clicked, the button Generate the Batch creates a new task, the page reloads to the task manager with an addition option in the pending selection list.

In depth explanations of the inputs will be provided in the VPP section of this report.

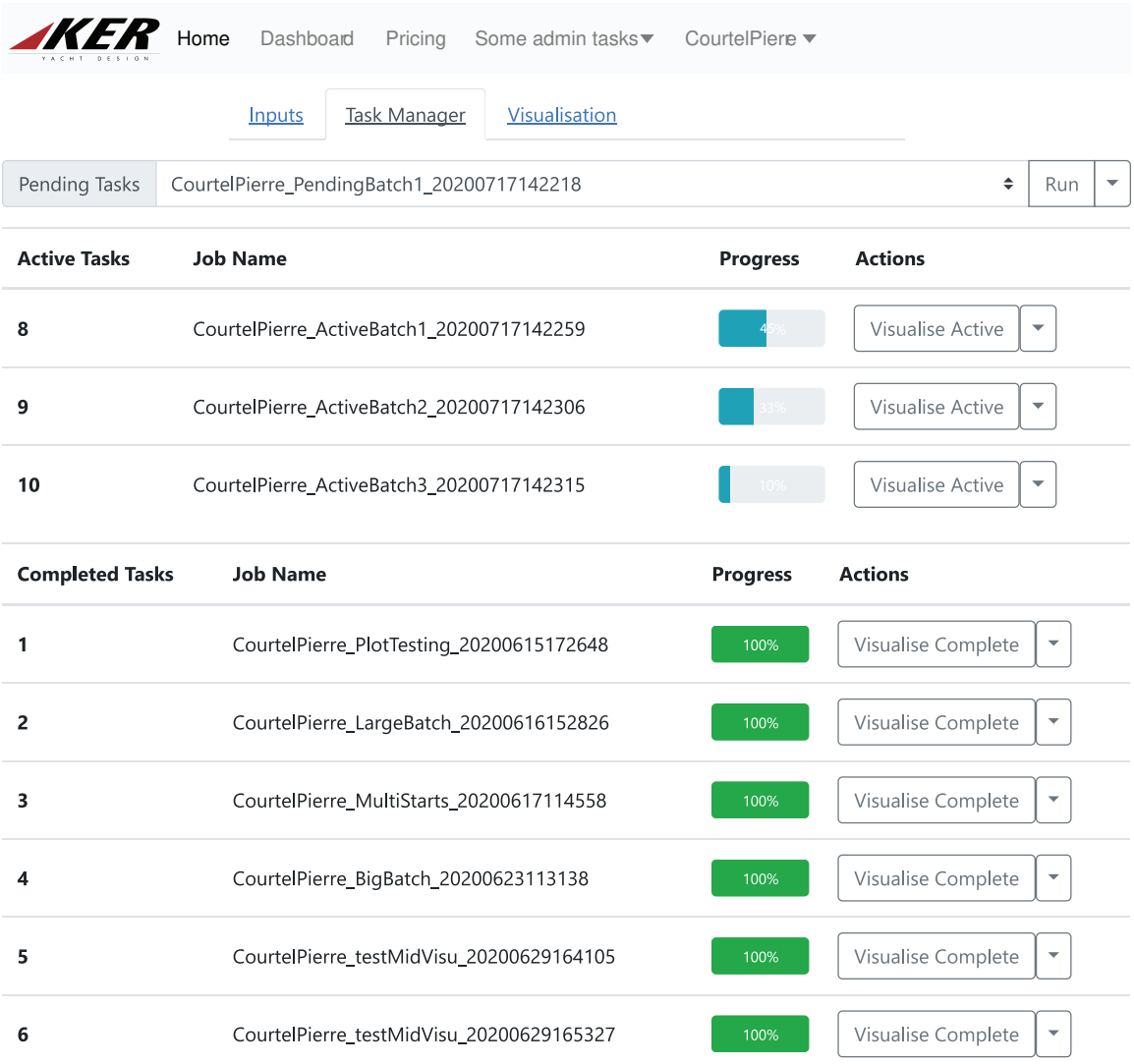

Figure 3.6: Task Manager

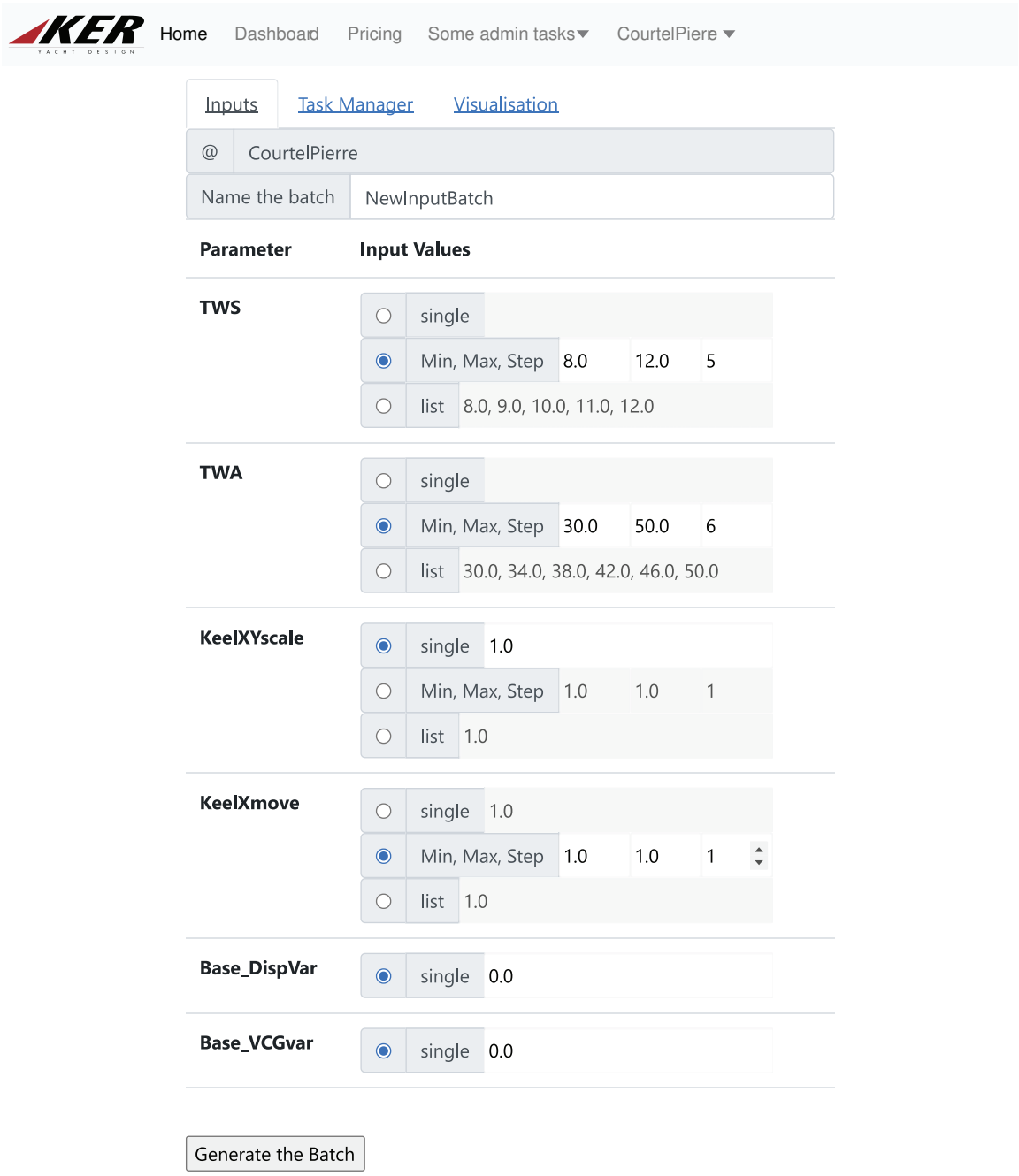

Figure 3.7: Dashboard input tab

3.4.7.3 Visualisation tab

Finally, the visualisation tab compiles the data into charts. The plot takes inspiration from the pivot chart available in Microsoft's Excel spreadsheets. The first top section of the page lists the filters that are applied to the figures. These filters can be handpicked for any of the parameters available in the VPP. The selection for these parameters is performed in the table at the bottom of the page. Here again additional explanations will follow in chapter 5, as this chapter focused on the website implementation only.

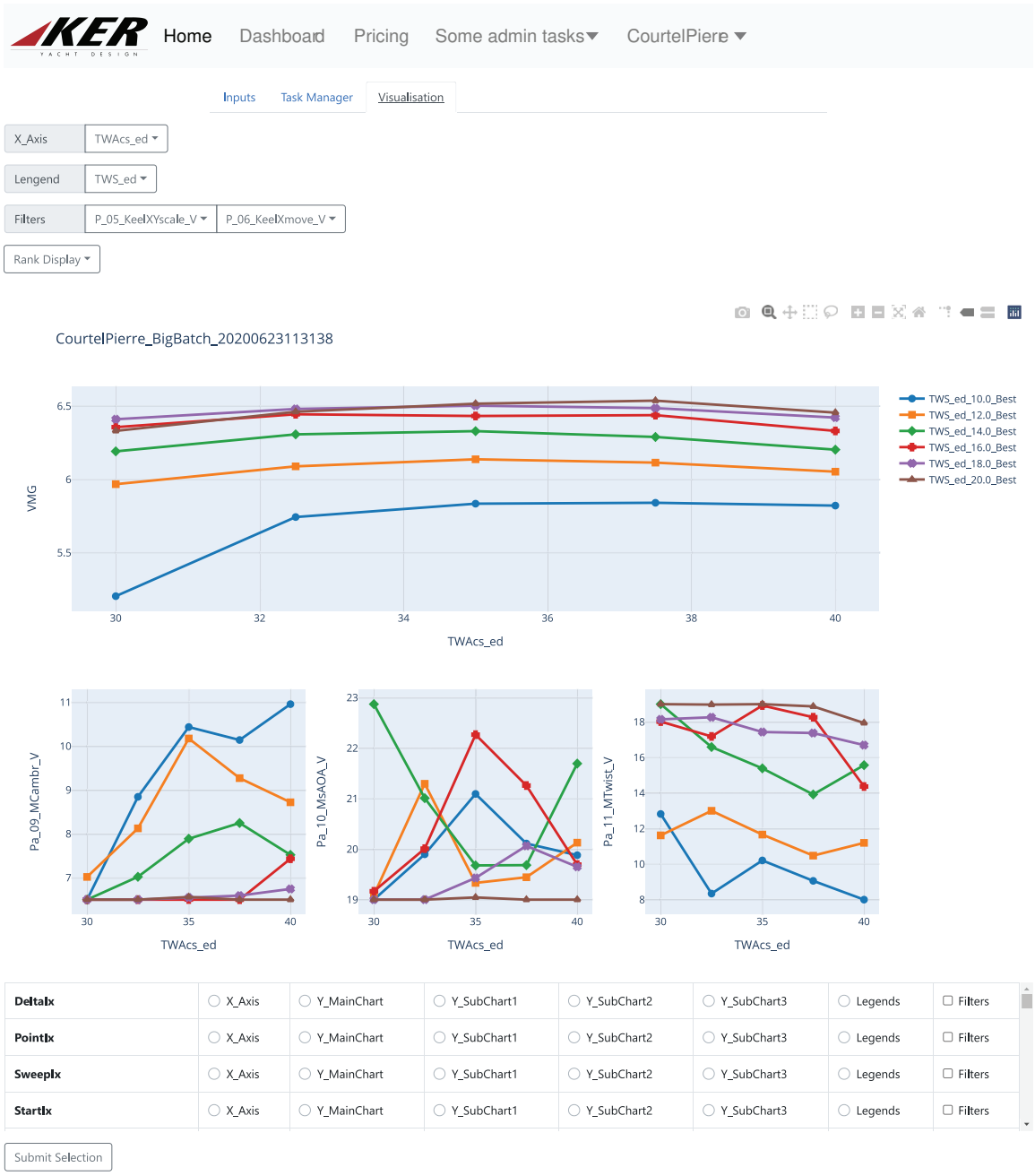

Figure 3.8: Dashboard Visualisation Tab

3.5 Integration of the VPP in the website

The python scripts used for the VPP and their integration to the website are described in fig 3.9.

The horizontal separations (User, Web Server, Cluster Master Node and Cluster Processing Node) correspond to the four computers in the system. The user and web server essential

to any online application have already been described in section 3.2. The cluster machines are the ones deployed on the cloud, Amazon Web Services in this particular instance.

The vertical subdivisions describe three actions performed on the website, the main one being the VPP system on the second line.

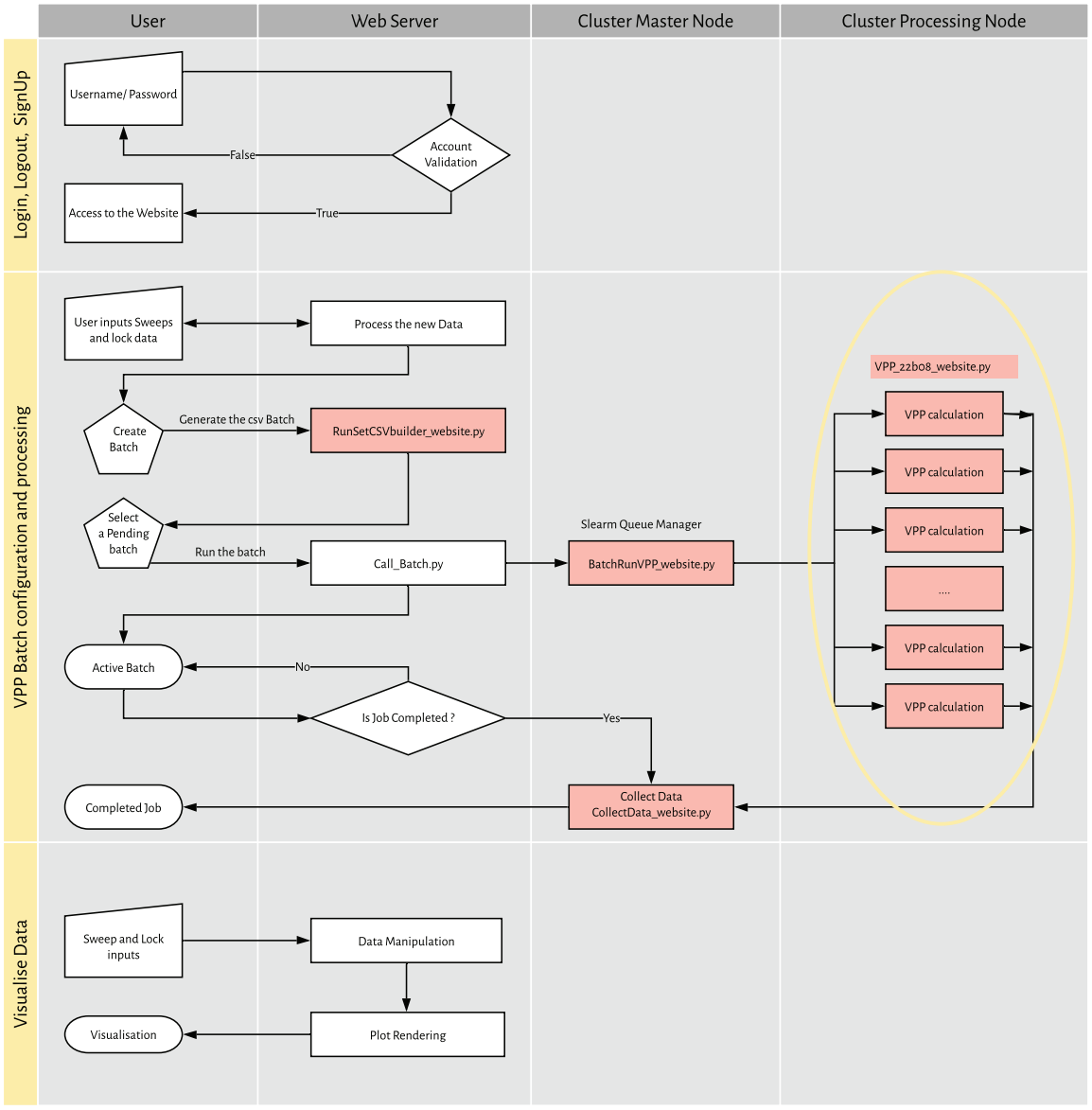

Figure 3.9: Process diagram describing the website and the VPP scripts

The VPP elements have been highlighted in red in the diagram. It is important to note that these scripts have been written at Ker Design before the internship of the author. Modifications have been implemented to these codes to enhance the performance, as discussed in section 4.2 but also minor modifications to adapt the scripts to the web frame.

The first process in the chain of operations takes care of the batch creation. To do so the data that has been given by the user, for example in fig 3.7, is processed by the script

`RunSetCSVbuilder_website.py`. Here a .csv file is generated gathering all the combinations of inputs. For the example given, the csv has 30 lines, corresponding to the combinations of the 5 True Wind Speeds (TWS) and the 6 True Wind Angles (TWA). For each line in the csv a VPP point will be calculated. This step can take a few seconds if the batch is composed of several sweeps but generally it is instantaneous.

With the csv generated, the batch acquires the pending status. To launch it on the cluster the script `BatchRunVPP_website.py` needs to be called on the master node. The role of this program is to create python subprocesses for each line in the csv. After this step a workload manager, the slurm queue, is put in place to optimise the computation. If the AWS cluster node has a processor of 16 cores, the queue is in charge of assigning subprocesses to each of those cores. It maintains the processor and replaces finished tasks with new ones. This ensures the workload is parallelised across the processing core capacity.

The subprocesses mentioned in the previous paragraph are represented on the chart with the VPP calculation boxes. These all run the same piece of code, `VPP_22b08_website.py`, with the various inputs from the initially generated csv. Each individual VPP script outputs a csv file which contains the data at equilibrium for a particular input combination. For the example in fig 3.7, the batch is expected to output 30 different csv files. This information is used to update the blue progress live in the task manager while the batch is computed (as seen in fig 3.6). An asynchronous request regularly runs a function which counts the number of files in the active batch output folder.

Once the batch is completed, the `CollectData_website.py` script is called. Its purpose is to gather the data of the individual csv generated by the multiple VPP scripts into a single file. The collected data csv are then used with the plotting function to display charts. This system offers one big advantage, because the data are generated pointwise it is possible to call the collection script mid way through the batch and plot the first points on the screen. Should the first points not be suitable it is possible to stop the computation and save precious processing time.

3.6 Dynamic Browser Charts

Websites have become more and more dynamic over the years, to the point that uploading a static graph has become out fashioned. The output charts visible on the website have been built around the *plolty* open source library. Originally coded in JavaScript then extended to python this package allows users to interact with the on-screen data as the mouse hovers on the curves.

With this tool the plots can be manipulated as if the user was plotting in his Python IDE of choice.

3.7 Data Storage and folder structure

The storage drive is composed of three main folders, the website, the VPP and the database as can be seen in fig 3.10. Both the server and the clutter machines will have access to the storage system.

3.8 Task Manager Database

A second table has been created in the database called job_mgmt. The intention is to save the information linked to tasks in the database. On project hard drive, the data for each task is saved in subfolders that have been named with hashes under *VPP Output Website* folder. In this way, it is possible to save all the tasks in one place regardless of which user is launching tasks. The database is then used to search the hashed folder corresponding to a given user name, job name and date. The fields contained in the new table are given in the table 3.2

Table 3.2: Description of columns in the SQL job_mgmt table.

Columns	Type
userId	Integer
userName	Characters
jobName	Characters
date	Date Time
foldeName	Hashed Characters
status	*Pending*, *Active* or *Completed*
priority	*low*, *normal* or *high*
cpuTime	Time
batchSize	Integer
currentSubPro	Integer

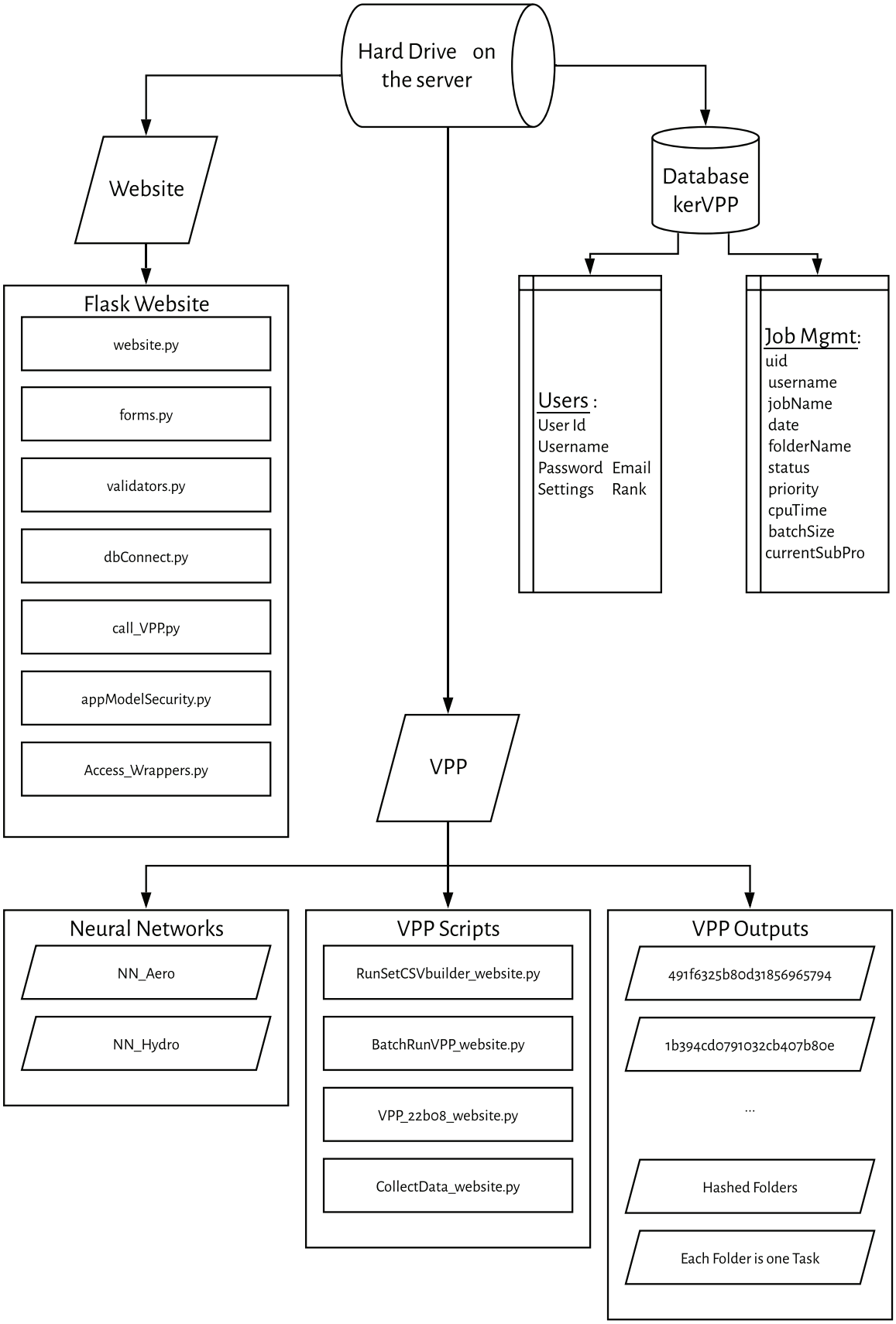

Figure 3.10: Storage structure diagram of the project.

3.9 Conclusion

With the website built, offering a functional VPP interface for cloud-based services a major goal of the project is achieved namely that the user management service and security systems put in place ensure safe access for multiple customers to the platform. The dynamic visualisation tab provides interactive solutions for designers to interact with the VPP data. The following section will present the VPP in detail and compare the forces predicted with the empirical models.

Chapter 4

PART 2: EXECUTION OF THIS VPP

In this chapter

4.1 The Neural Network models used in the VPP

The VPP developed at Ker Design and implemented for this project is composed of very complex subsystems. In this chapter the boat, the aerodynamic and hydrodynamic models are presented to better understand how the VPP works. A performance analysis of the python scripts is conducted with hopes to optimise the code and reduce the overall running time of the program. Finally, the aero- and hydro models are compared to other existing to validate the forces calculated by the VPP. The hydrodynamic model is tested against the empirical DSYHS method [7], while the aero is checked against the ORC VPP aero model [21].

4.1.1 Aerodynamic model

The aerodynamic surrogate model (based on RANS CFD data) implemented in this VPP features a multi dimension space amongst other inputs. To clarify on the vocabulary, sails are tightened between three attachment points, the head located up on the mast, the tack and the clew both found at the bottom of the sail. The tack is the fore point while the clew is the aft point of the sails. The bottom segment of the sail joining the clew to the tack is

called the foot, the fore segment linking the tack to the head is called the luff and the aft segment joining the clew to the head is the leech. These terms are depicted on fig 4.1.

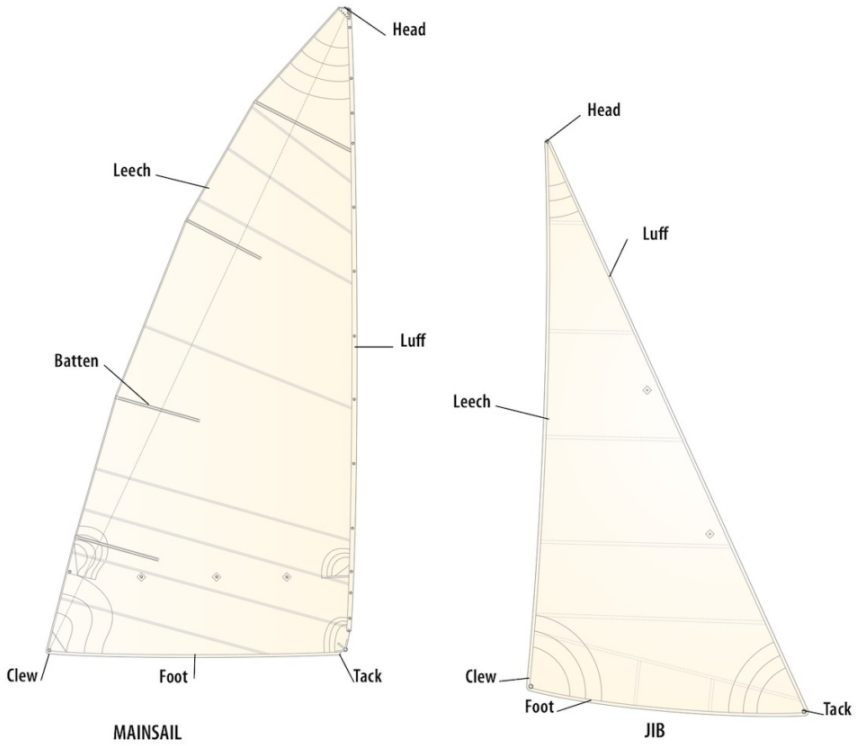

Figure 4.1: Vocabulary commonly used for sails [36].

The dimensions include key ratios between general rig and sail plan lengths. The fractionnality $(I/(P + BAS))$ gives the height ratio between the attachment point of the jib to and the overall height of the mast. P/E Describes the mainsail luff to foot ratio, I/J corresponds to the same ratio but this time for the headsail and MHB/E compares the head to foot ratio of the mainsail. The parameters (BAS, E, I, J and P) involved in those ratios are represented graphically in the fig 4.2.

Alongside these fundamental numbers, the ORC sail measurements are displayed in the figure as well. The ORC rules have defined a procedure to follow in order to determine the sail areas. For the main sails 5 horizontal measurements are taken. They are distributed along the luff of length P (main sail hoist) at respectively a quarter (*MQW*), a half (*MHW*), three quarters (*MTW*), seven eighth (*MUW*) and finally at the head of the sail (*MHB*). The head sail is defined with a similar method, the measurements are taken perpendicular to the luff (*HLU*) with the same distribution along the luff (1/4, 1/2, 3/4, 7/8, head). An additional measurement is required for the headsails that is (*HLP*) the smallest distance from the clew to the luff.

The AWA and heel are used describe the position of the yacht with respect to the wind environment.

Finally, the remaining dimensions model the sail shapes of the boat. The angle of attack and the camber are terms often linked with aeronautical foils and are used in similar context for the sails of the yacht. The twist and roach are specific terms to the sailing industry and will thus be explained further. To better understand these two parameters three views of the boat are given in the fig 4.3.

The twist of the sail marks the difference in the angle of incidence between the head and the foot of a sail. It has been highlighted on left image of the fig 4.3 with red lines marking the twist angle. Typical values for the twist can range 0 and 20 °.

The triangular shape created between the clew, the head and the tack forms the majority of the sail area. However, most sails have an additional surface aft of the head clew line. The roach characterises this specific area it can be observed for the main sail on the image situated on the middle of fig 4.3.

Rig	
P	15.025
BAS	1.585
E	5.88
I	15.115
J	4.84
Mainsail	
MHB	1.57
MUW	1.98
MTW	2.71
MHW	3.88
MQW	4.85
Headsail	
HHB	0.07
HUW	0.84
HTW	1.56
HHW	2.69
HQW	3.81
HLP	5.01
HLU	15.16

All units in meters

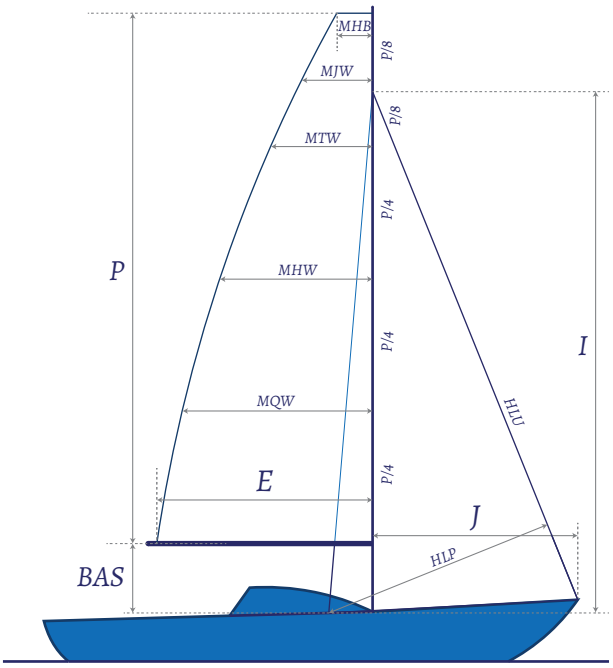

Figure 4.2: Sail plan dimensions inspired by North Sail [37] and redrawn by the author to display the ORC measurements.

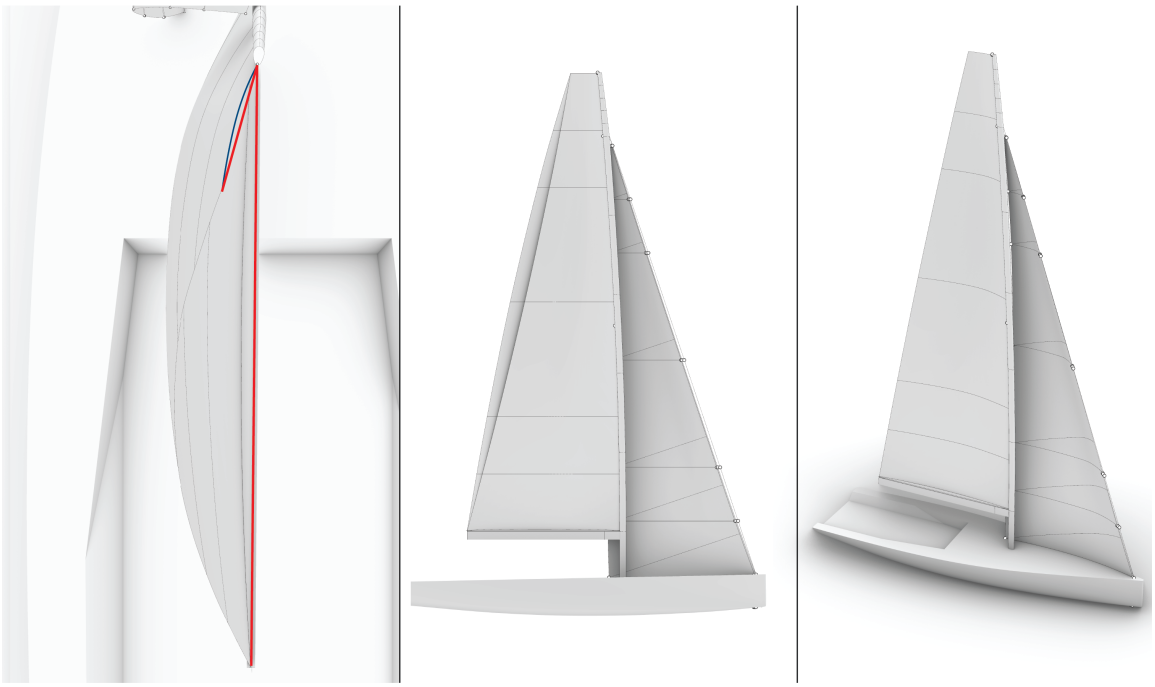

Figure 4.3: From left to right, the top view of the mainsail with the twist angle highlighted, the side view of the sail plan with the mainsail roach shaded, the 3d rendering of the yacht.

4.1.2 Hydrodynamic model

The hydrodynamic surrogate model (also derived from RANS CFD data) is composed of less dimensions with 8 parameters open to investigations as given in table 4.1. The first 6 parameters model the behaviour of the hull in the water, these are essential to find the hydrodynamic forces. The last two have been added to the VPP to assess the effect of the size and longitudinal position of the keel. The keel scale factor can reduce or enlarge the XY dimensions within a range of 0.8 to 1.2 while the position of the keel can be pushed up to 600mm aft or fore of its original position.

The choice for a CFD based model is mainly dictated by the precision required to accurately represent the behaviour of the yacht. Alternatives such as the DSYHS could be a good option at the preliminary level because it is very simple to compute the forces. The order of magnitude is of seconds whereas for the surrogate model presented here the CFD calculations can take several days to run on a dedicated server at Amazon.

Table 4.1: Description of the 8-dimension space used in the hydrodynamic surrogate model

Dimension	Description
Speed	Boat Speed
Heel	Heeling angle of the boat
Leeway	Leeway angle of the boat
FzH	Heave forces on the boat
MyH	Pitching moment
MzH	Yaw moment
KeelXYscale	Multiplication factor of the keel
KeelXmove	Longitudinal location of the keel

The hydrodynamic model presented here is based on a 40ft hull with its key characteristics given in the table 4.2 and a 3d view of the geometry tested in CFD in given on

Table 4.2: Main dimensions of the Hull

	Hull
Disp	6.8969 m^3
BWL	2.806 m
LWL	12.286 m
T	2.4124 m
T_C	0.5786 m
Cp	0.5309 -
Cb	0.3764 -

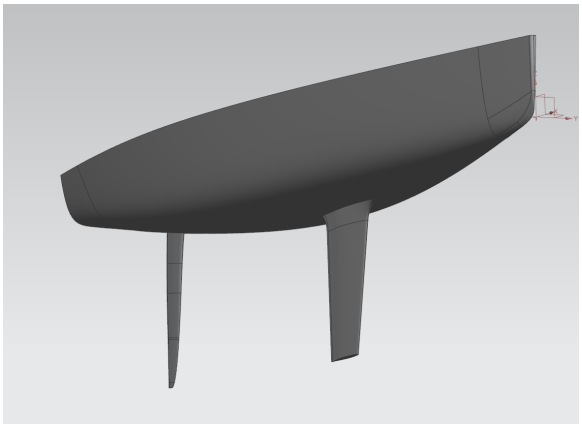

Figure 4.4: Geometry representation of the hull.

4.1.3 Neural Networks

With the multi dimension aero- and hydro models, the number of combinations to inquire on can get large very quickly. To limit the CFD costs Neural Networks (NNs) are used to interpolate new points within the CFD data set. The data set is divided in two unequal parts, the larger set is used to train the network while a smaller one is kept to test the network. This approach allows to monitor the quality of the network with an R-squared error when the small set is tested.

As with any interpolation method overfitting can occur, in this case the function is accurate in close range to the initial data points but then spikes out of control in between the initial points. To mitigate this problem the model relies on a set of networks (20).When inquired for new data point the networks are ranked based on their error. Once ranked the output data is calculated with an averaging function taking into account only the best networks (3 to 5) . With this method the risk of running into numerical errors is controlled.

With the neural networks built and trained the VPP algorithm can start the balance procedure. A minimising function is used to solve for equilibrium. The goal is to find a combination within the parameter dimensions that will balance the aero and hydro models while keeping the boat stable. All the parameters in the models have been constrained to range of acceptable values, this range is bounded by the limits of the CFD data set.

The number of distributed points is controlled with a sequencing procedure. A maximum number of points is set to 100 this number can be changed with ease. A simple option would be to distribute all the points in the full range of the circa 20 dimensions and find the best combinations. To improve the accuracy another approach is implemented. A first set of 60 points is solved, the solutions are ranked and based on the best solutions the field of search is narrowed downed by limiting the range of the parameters. Within this new constrained domain, a second set of 30 points is created and solved. The same procedure is repeated zooming further towards the optimum solution and the remaining 10 points are solved.

Going back to the input example given in the fig 3.7 the batch will have to compute 30 VPP instances. With the sequencing algorithm each of these 30 instances will have 100 configurations to solve, once solved the data is ranked based on the boat speed to find the fastest configurations. The data displayed on screen corresponds to the best rank by default but an optional drop down menu allows to display the 5 best configurations. Such a comparison can be seen in the fig 4.5.

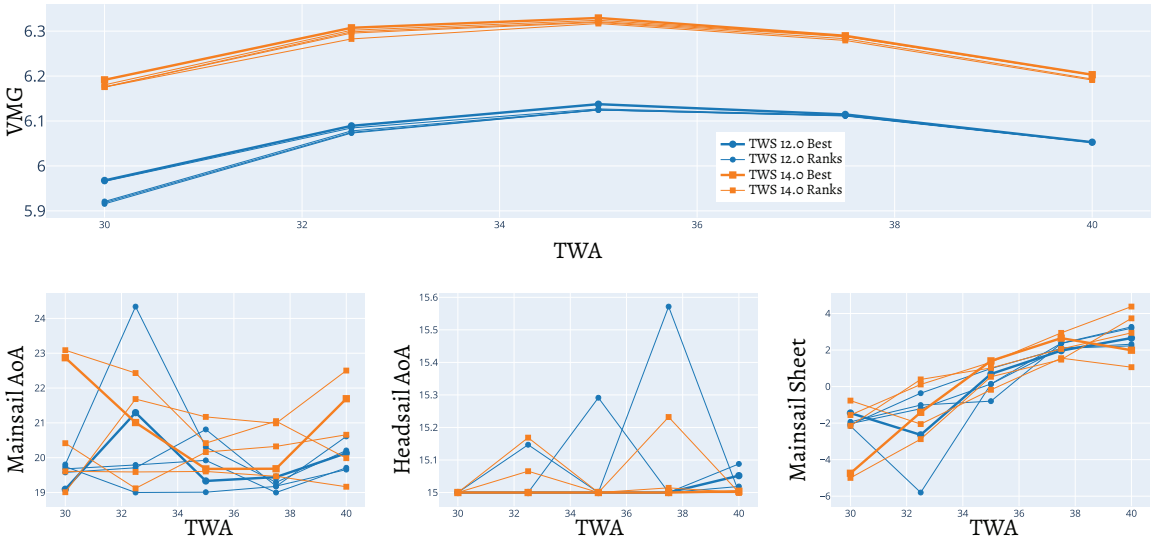

Figure 4.5: Comparison of two TWS speeds with their 5 best ranked solutions.

4.2 Performance study of python scripts

The running time to compute one VPP point needs to be as small as possible in order to have a competitive software. This section will look into performance issues seen with the VPP python framework in place at Ker Design.

4.2.1 The Downside to Using Python

Python is a popular programming language within the engineering community. It was designed at its core to be easily readable, traditional brackets and semi columns are nowhere to be seen. At times when other languages use punctuation, python uses English words providing a clutter-free code. Python also contains an automatic memory management system which saves a lot of coding when compared to C, furthermore, python uses types but the variables themselves don't have to be typed in the code. These features although very useful come at the expense of slower running codes. Overheads can be significant in python when for example calling functions or on attribute lookup.

The core functionalities of the language are intentionally kept to a relatively small number but a large number of modules can be imported to extend the language to suit particular programming needs.

4.2.2 Can C-like modules such as Cython speed up the VPP ?

With the constant growth of the python user base in the scientific community [38], needs to enhance its performance have also increased. Stefan Behnel is one of the developers providing a new tool, Cython, to write fast python scripts [39]. The idea is to compile python scripts into C to allow explicit type declaration in a Python like syntax, thus speeding up the code.

The procedure is very simple, the compiler can transform any python code in C, thus writing thousands lines of C code automatically. Only marginal gains have been observed when compiling the VPP scripts, with running times in the order of 1% faster. Furthermore, with the additional time (the order of magnitude is around 5 minutes for the VPP script) taken to compile the code it is not yet worth it. To really make use of Cython the programmer needs to explicitly type variables in the code. Doing so the compiler can remove most of the python overhead verification's on typed variables speeding the code. At that stage the additional compilation time is worth it as the codes can be up to a hundred times faster to run.

To present the cython language the listing 4.1 gives examples of a simple snippet of code written in both python and cython. The explicitly typed script remains very similar to the original version.

Listing 4.1: Comparison of the same simple code written first in plain python then additional explicit typing to turn it into faster Cython

```
####################
# example_original.py
####################
from math import sin

def f(x):
    return sin(x**2)

def integrate_f(a, b, N):
    s = 0
    dx = (b-a)/N
    for i in range(N):
    s += f(a+i*dx)
    return s * dx
```

```
############################################
# example_cython.pyx, Explicit typing is added
############################################
from math import sin

def f(double x):
    return sin(x**2)

def integrate_f(double a, double b, int N):
    cdef int i
    cdef double s, dx
    s = 0
    dx = (b-a)/N
    for i in range(N):
    s += f(a+i*dx)
    return s * dx
```

```
##################################################
# Even_faster_cython.pyx, C library "math.h" is added
##################################################
cdef extern from "math.h":
    double sin(double)

cdef double f(double x):
    return sin(x*x)

def integrate_f(double a, double b, int N):
    cdef int i
    cdef double s, dx
    s = 0
    dx = (b-a)/N
    for i in range(N):
    s += f(a+i*dx)
    return s * dx
```

The programming tutorials presented by Behnel [40] where the previous snippets have been extracted show clear running time improvements. Four cases are outlined in the paper. The original python which serves as the reference time. The compiled python, without explicit typing, which runs 5% faster, the compiled python with explicit which runs 24 times faster and finally the compiled python with explicit typing associated to faster C libraries with the time being reduced by a factor 219. The objective is to apply the method to the VPP, find the bottleneck variables and explicitly type them to increase the performance without having to rewrite the entire script.

Although very promising this package proved not to be useful in the context of this VPP. Indeed the profiling study performed in section 4.2.3 showed the limiting factor of the application to be the PyTorch Neural Network library. Typing the tensor variables is unfortunately not possible and no significant gains have been found when compiling the python scripts without explicit typing. Other options could be investigated in the future if the speed is required to increase drastically. The PyTorch library also comes in a C++ API therefore some parts of the VPP could be rewritten in C and called from the python website framework but this outside of the scope of this master thesis work.

4.2.3 Profiling the VPP

When it comes to code optimisation the first action to take is to evaluate the code with a profiler. The profiler runs in the background when the script is launched and measures the time taken by each individual function. With this information it is possible to track down the bottlenecks in the code and make modifications to the slowest elements first.

Script profiles taken optimise the code have shown a bottleneck when manipulating data with the pandas library. A modification in the code replacing the slow pandas dataframe with faster numpy array reduced the running time by a factor 7.

Additionally the new version script has shown that the limiting factor of the application is now the neural network inquiries. As mentioned in the section 4.1.3 set of neural networks are averaged to prevent numerical errors. This averaging function alone represents 63% of the overall running time of the VPP. The improvements made in the code will allow to run the VPP considerably faster.

4.3 Validation of the VPP Forces

This section will compare the VPP forces with other models to establish whether the results are coherent. The hydrodynamic comparison model is built on the Delft Systematic Yacht Hull Series as originally developed in the 1970s by Gerritsma [42] and continued by Keuning et al. [7]; the various elements have been compiled by Souppez [43] and are detailed in the appendix A.

4.3.1 Forces Given by the VPP

Using the website it is possible to change the axis of the charts on display. The fig 4.6 contains the data required to compare the VPP to the DSYHS model, the boat speed, the heeling angles, the longitudinal hydrodynamic forces and the aerodynamic side forces.

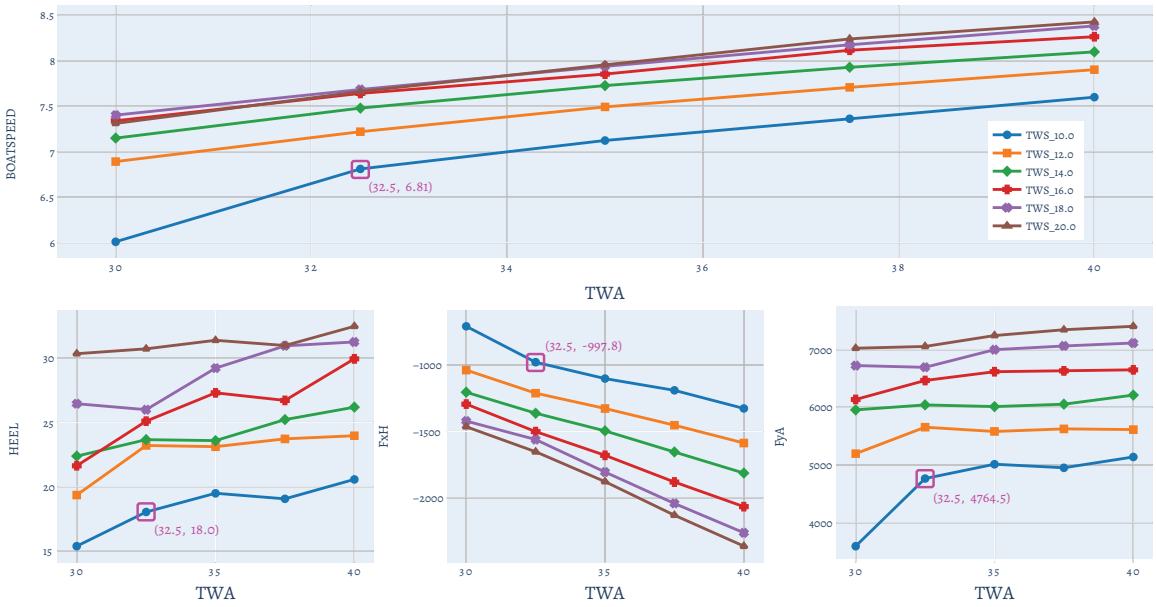

Figure 4.6: Website charts export displaying the Boatspeed [kts], the heel angle [°], the longitudinal hydrodynamic force FxH [N] and the aero dynamic side force FyA [N]. One of the points is highlighted in pink and its values are explicitly written to ease the comparison.

4.3.2 Delft Series Hydrodynamic Model

The procedure followed to build the forces of this DSYHS model has been completely described in the appendix A. The results are displayed in the fig 4.7 and the two models are well correlated. One point VPP point was explicitly highlighted in pink in the fig 4.6, because of the website implementation it is not possible to change the axis to plot the forces as a function of the boat speed. The heel angle, boat speed and longitudinal force (18°, 6.81 kts, -997.8 N) must be taken from the y-axis of the three respective subplots. This same point has been reproduced on the DSYHS figure.

The DSYHS force curves are plotted for four heeling angles 0°, 10°, 18° and 30°, doing so other points in the VPP can be checked on the same graph. For angles in between those four the method simply uses a linear interpolation. The pink point on the DSYHS graph is placed with a boat speed of 6.8 knots and the heeling angle at 18 ° the resulting force is just above a 1000 N.

To make sure this is not a coincidence on this particular point additional configurations can be checked, for a TWS of 14 knots and a TWA of 37.5° the boat speed, heeling angle and resistance are respectively 25.2°, 7.93 knots and -1654 N. When in use on the website the VPP plots feature dynamic data reads. When the mouse hovers on a data point, the exact values are displayed in a small pop up notification box. For this second point, the DSYHS model predicts a force around 1750 N which is a bit higher than what the VPP is showing but still remains acceptable and coherent.

Overall when looking at the forces in both models for boat speeds between 6 and 8 knots and heeling angles from 15 to 30 °, the range of forces corresponds building confidence in the VPP hydrodynamic model. With this validation procedure, the order of magnitude of the hydrodynamic forces has been checked, the next action to take is to access the aerodynamic forces.

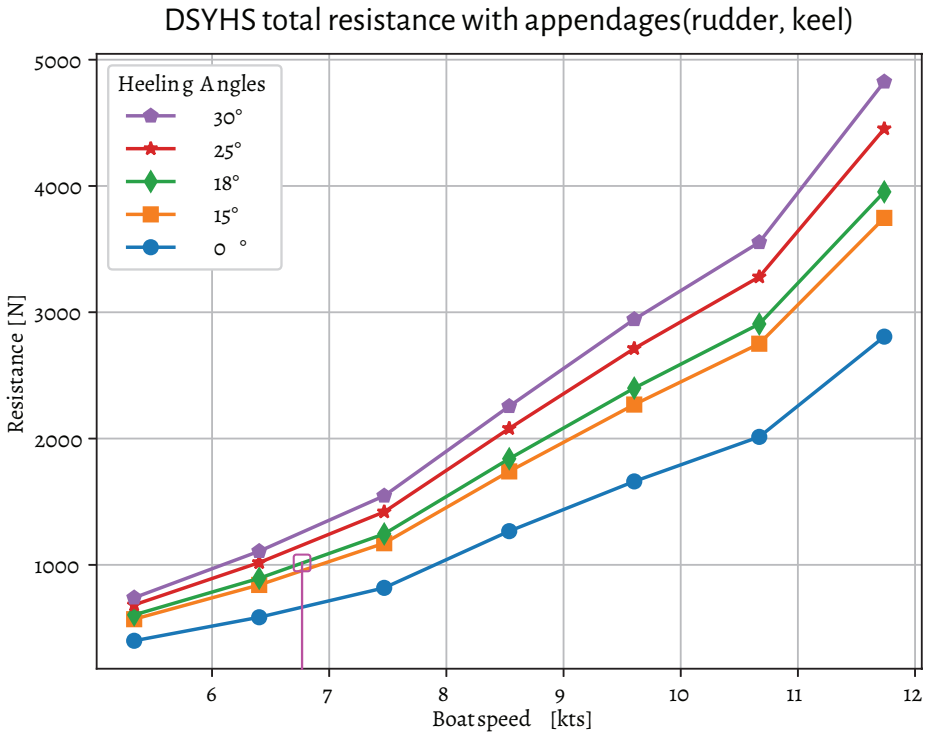

Figure 4.7: Hydrodynamic forces of the appended hull computed with the DSYHS regression model.

4.3.3 ORC aerodynamic model

Having evaluated the hydrodynamic model of the VPP against the DSYHS method, it is now the time to review the aerodynamic model. The comparison is performed with the ORC model used in their VPP to rate competitors yachts in regattas.

The empirical methodology followed to determine the loads were developed by Hazen in 1980 [26] and is amongst the favourite within the industry. The ORC has published a set of lift and drag coefficients that are used to determine the forces acting on the sail plan, they can be found on fig 4.8. On this figure the coefficients for the AWA of 18° have been highlighted, this angle has been taken as it corresponds to the point that was tested in the DSHYS method in the previous section. Furthermore the apparent wind speed for that same point is 16.16 knots.

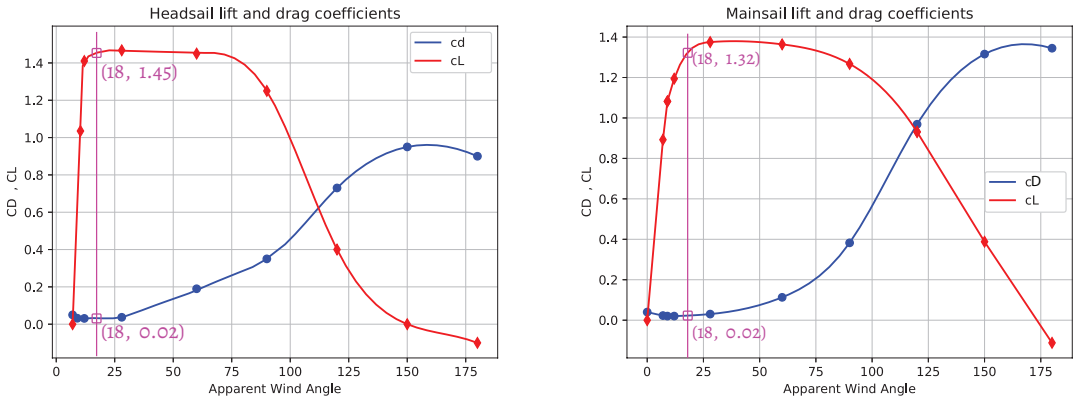

Figure 4.8: ORC lift and drag coefficients for both the head and main sail interpolated from the values given the ORC VPP[21].

The drag and lift coefficients that are given for the head and main sail need to be regrouped to form coefficients for the two sails. The following equations are used for this purpose,

$$C_L = \frac{C_{L\,main} \cdot A_{N\,main} + C_{L\,head} \cdot A_{head}}{A_{N\,main} + A_{N\,head}} \,. \tag{4.1}$$

Where the areas are given as functions of the sail plan dimensions presented on fig 4.2

$$\begin{aligned} A_{N\,main} &= 0.5 \cdot P \cdot E \,, \\ A_{N\,head} &= 0.5 \cdot I \cdot J \,, \\ A_{head} &= 0.5 \cdot \sqrt{I^2 + J^2} \cdot HLP \,. \end{aligned}$$

The same equation is used to determine the parasitic component of the drag:

$$C_{Dp} = \frac{C_{D\,main} \cdot A_{N\,main} + C_{D\,head} \cdot A_{head}}{A_{N\,main} + A_{N\,head}} . \tag{4.2}$$

The total drag coefficient is composed of two additional entities. The lift-induced drag and the windage drag which considers the effects of the Rig and Hull. The former is defined as:

$$C_{Di} = C_L^2 \cdot \left(\frac{1}{\pi \cdot AR} + 0.005 \right) . \tag{4.3}$$

Here the additional term 0.005 models the drag caused by the flow separation and the aspect ratio AR is taken according to the close hauled (upwind sailing) scenario given by Hazen:

$$AR_{ch} = \frac{\left(1.1 \cdot (EHM + FA)^2\right)}{A_N} . \tag{4.4}$$

Where EHM the mast effective height is taken as $(P + BAS)$ and FA is the average freeboard of the hull.

The third and last component of the drag coefficient takes the rig into account, the windage is estimated as:

$$C_{D0} = 1.13 \cdot \frac{BOA \cdot FA + EHM \cdot \phi_{mast}}{A_N} . \tag{4.5}$$

With BOA the beam overall and ϕ_{mast} the average diameter of the mast.

The total drag coefficient is therefore

$$C_D = C_{Dp} + C_{Di} + C_{D0} \tag{4.6}$$

The lift and drag forces can be calculated thanks to these coefficients in the following equations:

$$L = \frac{1}{2} \rho_{air} \cdot A_{sails} \cdot AWS^2 \cdot C_L \tag{4.7}$$

$$D = \frac{1}{2} \rho_{air} \cdot A_{sails} \cdot AWS^2 \cdot C_D \tag{4.8}$$

$$\tag{4.9}$$

The lift and drag force are respectively acting, in the sail frame of reference, perpendicularly and along the sails. One last calculation is required to compute the Drive and Sail side force. The diagram given in the fig 4.9 describes the sailing equilibrium of the forces in action.

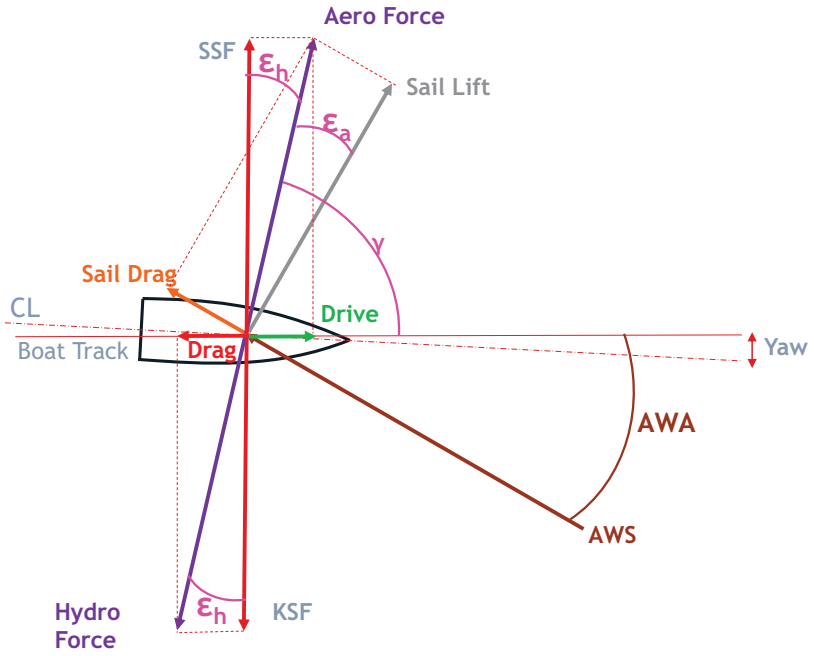

Figure 4.9: Sailing equilibrium diagram [44]

Finally, the lift and drag forces are projected in the frame of reference of the yacht, the drive and sail side forces are given by the following equations:

$$F_D = L\sin(AWA) - D\cos(AWA) \tag{4.10}$$

$$SSF = L\cos(AWA) + D\sin(AWA) \tag{4.11}$$

The method was implemented in the python script given in the appendix B, and results to a driving force of 907 N and a side force 4679 N. These forces are a close the ones outputted from the VPP, the drive force expected was 997 N while the side force was 4764N. These differences can be explained by the empirical nature of the model developed for this validation. Although very convenient to use because of their simplicity, the Hazen calculations cannot represent the full complexity of the with eleven equations. Nonetheless, knowing the limitations of the method, the driving force is under predicted by 10% while the side force is accurately modelled with only 1% difference, overall the order of magnitude is respected.

4.4 Conclusion

The validation models presented and developed throughout this chapter have shown that the VPP and empirical forces (for two degrees of freedom) are within the same order of magnitude. Although promising the validation at this stage of the VPP website can only be partial.

Indeed to better model the system additional CFD campaigns must be conducted, ideally for yachts where on water data is readily available to compare the numerical prediction to real racing conditions. Still under development the underlying VPP is in constant change and further detailed validation is for now premature the priority being placed on developing the website. The following chapter presents a concrete design scenario using the VPP to assess the performance of different design options.

Chapter 5

PART 3: RESULTS AND OPTIMISATION USE CASE

In this chapter

Part one of this report was focused on the development of the new cloud base interface presenting the functionalities implemented on the platform. The second part described the models implement in the VPP providing a validation between the hydrodynamic surrogate model and the DSYHS empirical regression model asserting the quality of the VPP. A performance study was also conducted leading to faster running times. This third part will present how the VPP can be used by the designer to make the right decisions.

5.1 Creation of a Batch

The first action to take when launching a new VPP instance is to create a batch. On the website this corresponds to the input tab of the dashboard described on the fig 3.7. The current set up offers the possibility evaluate six physical dimensions. The wind conditions are imposed with the true wind speed and true wind angle. It is also possible to study changes on the keel such as scale its size or moving the longitudinal position on the waterline. Finally, the displacement and the vertical centre of gravity can be modified.

These parameters are fixed for a given customer requirements. It could be possible to investigate other dimensions if needed but it requires an administrator to make modifications in the VPP code. The parameters can be injected in the VPP by one of three ways, a

single value, a linearly spaced vector created from a minimum, a maximum and the number of points in between or a manual list of values separated by commas. These three options are not necessarily displayed on the screen. On the fig 5.1 the entries *Base_DispVar* and *Base_VCGvar* have been limited to single values only while the four other parameters have the choice between the three forms.

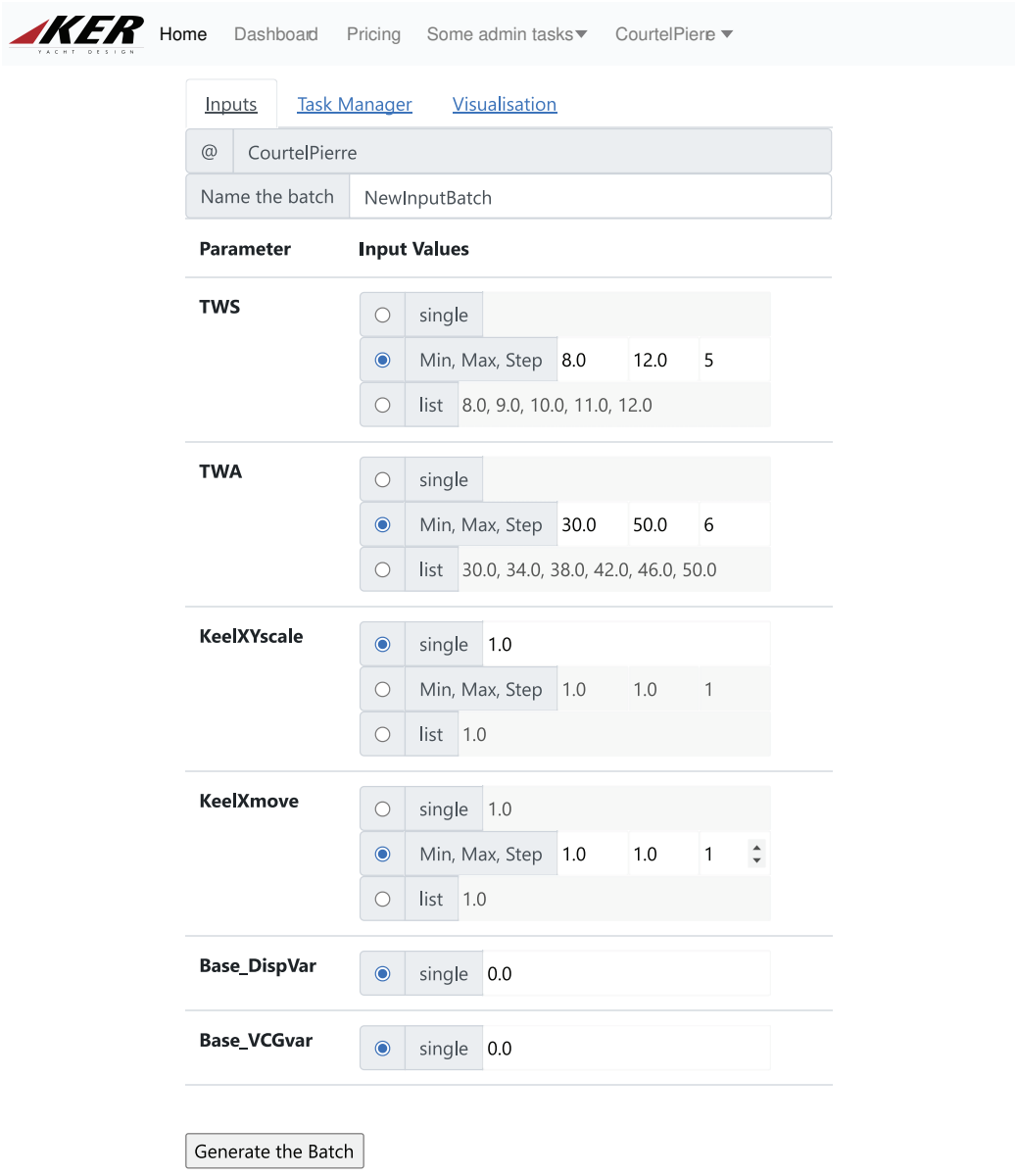

Figure 5.1: Website input tab,

5.2 Visualisation of the batch

With the batch created and completed the visualisation of the data is the last step of the process. The charts displayed on screen are modular, the user can change the data for each

axis and select filters to apply. Still under development this feature proves to be problematic as some combinations are not compatible on display. For a future customer use, the table would be reduced to a very limited set of options to choose from.

5.2.1 Selection menus on the legend and the x-axis.

The default parameter on the x-axis is the TWA, its unique values are gathered in a drop-down selection menu visible on fig 5.2. On this example the data points with a TWA at 32.5° have been removed. A similar menu is also available for the TWS legends. The batch in the same figure sweeps on the TWS from 10 to 20 knots with 6 points. When the legend selection only takes one of those 6 points, it is possible to display the best ranks leading to the maximum Velocity Made Good (VMG, the speed component in the direction of the wind).

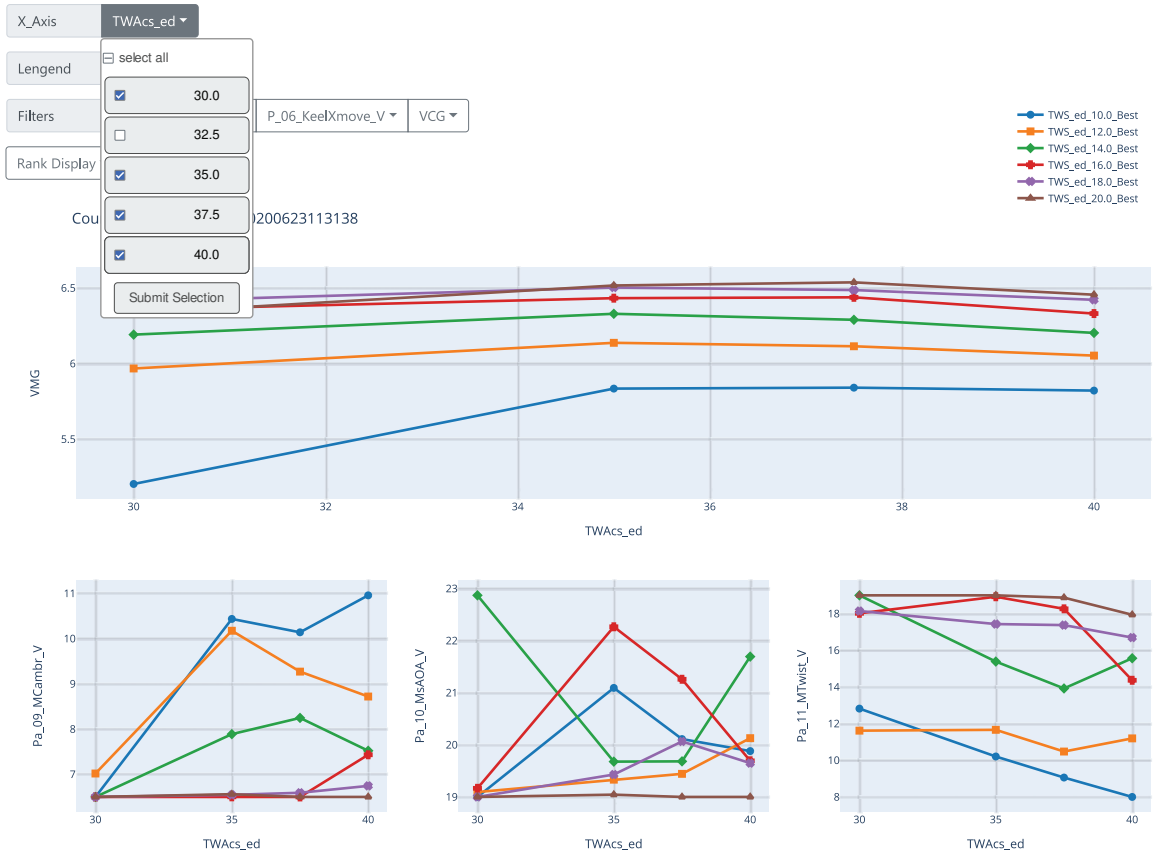

Figure 5.2: Filtering menus for the visualisation charts.

The ranks configuration is visible on fig 5.3. As can be seen on the figure, the secondary charts have been changed to show the Main and Head sail angles of attack as well as the

Main Sail sheet. The 5 best ranks reach similar VMG results, however, the parameters leading to them can change a lot. The mainsail AoAs are distributed between 19 and 23 ° meaning there are multiple trim settings available on the water. With the main sheet displayed on the bottom right figure, the trend shown is to ease the sheet while bearing away from the wind. On the other hand, the headsail AoA is fixed at the lower boundary of 15 °.

5.2.2 Filters

The filters are used to switch between configurations when parameters other than the wind-related ones are swept. The simple case presented in the fig 3.7 only takes single values therefore no options are available but another batch named as *CourtelPierre_BigBatch_20200623113138* displayed on figs 5.2 and 5.3 swept through 5 different VCGs. Radio buttons allow to select the desired configuration, originally selected to -0.545 it has been changed to -0.145 as shown on the fig 5.4.

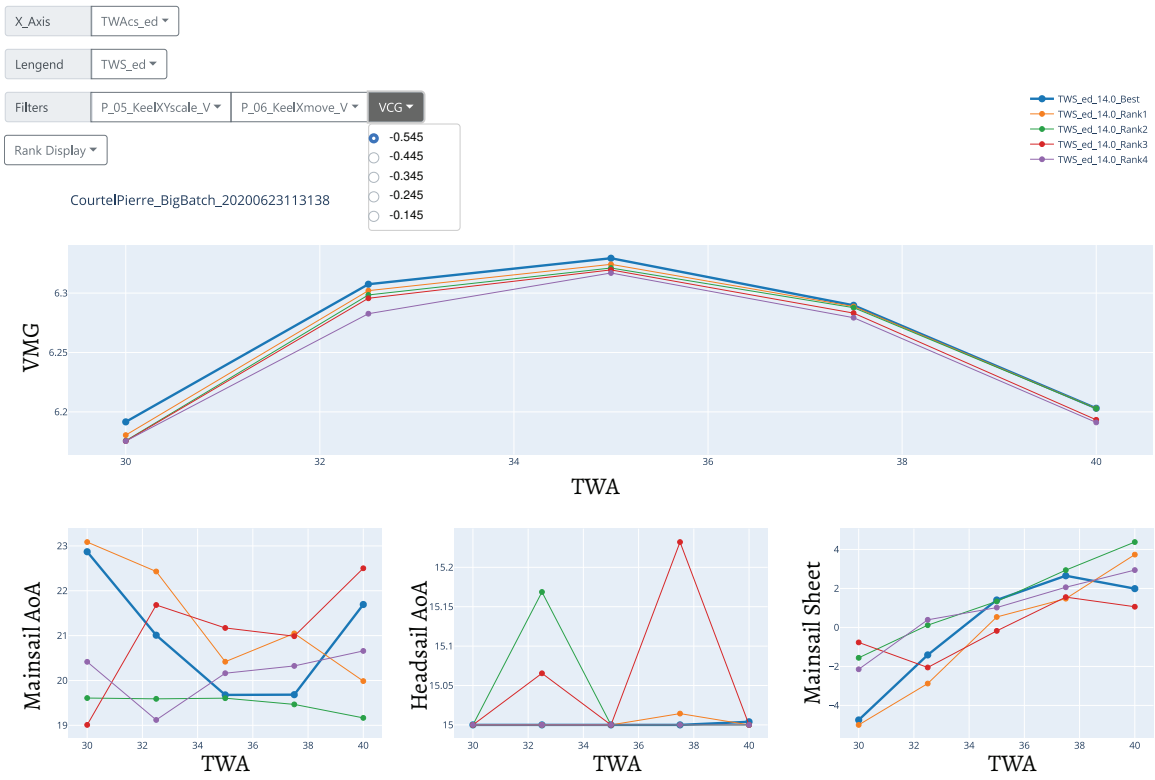

Figure 5.3: The legend selection is set to 14 knots and the 5 best ranks are shown.

The filters section on this figure shows the vertical centre of gravity (VCG) sweep performed. In this particular case with -0.545 being selected. This sweep was put in place to conduct the design study of a new keel for yacht owner willing to improve his boat. The VPP is used by the naval architects to compare the performance between various designs.

In this context the vertical centre of gravity of the boat, the longitudinal positions and a scaling factor of the keel have been observed.

5.3 Result Comparison

The position of the VCG can be changed with the mass of the keel, increasing it will lower the VCG. Between the two figures presented (figs5.3 and 5.4), the VMG is dropping significantly with speeds being reduced by about 0.3 knots throughout the TWA range. These changes can be significant while sailing on a race course. To explain this drop the physics behind the problem must be defined. Starting from the heavy keel scenario reducing its mass will lower the righting moment of yachts. For the same wind conditions and heeling moment, the vessel is no longer balanced and the heel increases. Additional heel implies a depowering of the sails which in turn decreases the driving force and speed of the boat.

The VMG data for the two cases have been summarised in the table 5.1. For the upwind conditions presented here, the VMG is increased for all the angles considered. The average progression is of 4% with a spike reaching close to 5%.

Table 5.1: Comparison of the VMG for the two extremes of the VCG Sweep

TWA	30	32.5	35	37.5	40	
VMG	5.90	6.04	6.06	6.08	6.02	VCG = -0.145
	6.19	6.30	6.32	6.28	6.20	VCG = -0.545
	+4.91%	+4.30%	+4.29%	+3.28%	+2.99%	

For the sweep considered here the lowest VCG leads to the largest VMG, this is the case if the vertical center of grivity of the boat can be lowered without changing the hydrodynamics of the hull i.e. switching to lighter carbon rig, or sails.

The output parameters generated by the VPP serve another key purpose for the designer. With boat refits where customers make a priority to raise the performance of their yacht while in regattas, the faster configuration on the water does not necessarily lead to more race wins. The penalty systems put in place by race organisers such as the ORC [21] to level out the field must be taken into account. These outputs can be fed through the rating system adding a second layer of complexity for the boat designer.

Finally, the VPP currently presented is based on CFD measurements performed in flat water, the wave modelling is therefore not considered. The keel position can have a big impact

on the performance in waves. The pitching motion of the yacht associated to waves often leads to the bow of the boat having to pierce through waves and in some cases the bow can be submerged. These phenomenons are not anticipated by the VPP so the results need to be carefully analysed as on water performance can differ from the numerical prediction.

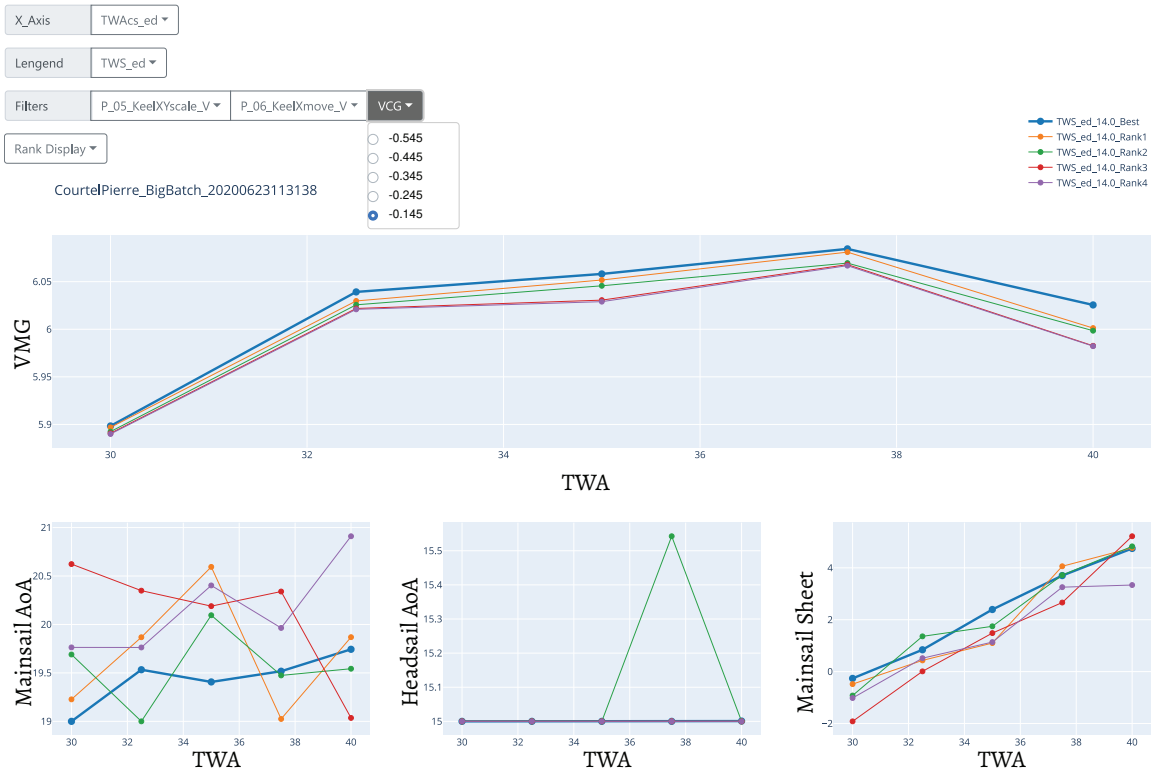

Figure 5.4: The legend selection is set to 14 knots and the 5 best ranks are shown, VCG -0.145.

5.4 Conclusion

The results presented here have shown one of the many procedures boat designers can employ to find performance gains for yacht owners. With this system it is possible to make simple changes to parameters and have a direct visualisation of the results on the boat speed (or on the VMG). Although representative of the real world behaviours, the speeds obtained must always be questioned as the waves are not considered in the hydrodynamic model. An experienced user is required to know how to read through the data presented on the website.

Chapter 6

CONCLUSION

6.1 A novel online application

Extensive programming has been performed to create a new interactive cloud based VPP solution. Building on the knowledge acquired by the engineers at Ker Design with years of experience in the yachting industry, this website opens the way for new customers to access the latest CFD technologies. Thanks to the economies of scale found by the giant cloud companies, the costs of outsourcing computer calculations are now very competitive. The website built in the context of this project can generate a new economical market for design offices allowing them to sell their complex technologies as a product for online customers.

On the platform the users have tools at their disposition to create, manage and visualise VPP data. During the creation phase the users are authorised to choose, modify and adapt a selected set of input parameters before launching the calculations. The management section of the website keeps track of tasks associated to the user, active tasks possess live updates informing the client of the calculation progress. Finally, interactive charts are available to display the data with options to filter and select certain information.

As for any application developed for web usage, the security has been a fundamental part of the project. The user system in place restricts the access to the platform to registered customers only. Common account functions to log in/out, sign up, have been implemented successfully. The password for the account is hashed and saved in a dedicated database, appropriate validation is implemented to prevent very simple passwords from brute force attacks.

Both the aerodynamic and hydrodynamic surrogate models built at the core of the VPP have been tested against empirical models with good results. Although the DSYHS method has been found to match the hydrodynamic model, the Hazen calculations based on the ORC lift and drag coefficients resulted in differences of up to 10%. Nevertheless given the empirical nature the test models and taking their limitations into account the models are

found to have corresponding order of magnitude for the two degrees of freedom tested. Finally, a use case for the VPP was presented, to investigate the effects associated to changing the mass of the keel a range of vertical centres of gravity have been taken as inputs. The VPP predicted a VMG speed increase of around 5%.

6.2 Further Work

Initially intended for a sail maker willing to have access to a VPP in order to test sail shapes for various hulls additional work remains to fulfil this desire. The VPP in its current state is set up to work with a predefined hull shape. In practice, changing the hull requires building a new geometry followed by extensive CFD calculations used to train the neural networks. In order to automate the process building a parametric hull and adding those parameters as new dimensions to the neural network model could be a way to proceed. Another solution at the preliminary stage of design could be to implement the DSYHS regression model to get results in a considerable shorter time frame.

At this stage the application is still in production and therefore the website has not yet been deployed the amazon cloud. This critical step in the life a website is often the source for bugs, as the production and deployment sometimes run on different operating systems. Particular attention was given in the proclamation phase to have the system working both in Windows and Linux seamlessly. Once online websites are constantly attacked by malicious scripts looking to access the data. As a safety measure at first and because for the time only a handful of customers are using the platform, it is probably best to filter out the connections to only allow selected IP addresses to connect. Adding another layer on top of the log in requirements.

Before the deployment small changes are required on the password system. Indeed the system in place to change the password imposes a manual action from the administrator to the password database. This is less than ideal and a proper password recovery system can be implemented without compromising the security of the platform.

Account privileges on the site are for now limited to two states, logged in or logged out. Additional accounts could be implemented to filter the options displayed on the screen. The idea would be to add to the logged in status whether the account correspond to an end user, a technician, or an administrator. In the hierarchy, the administrator has full access to the system and can choose to give the technician controls on certain options of the system. The technician is then in charge of setting up the website for the end user.

References

[1] F. Fossati. "Aero-Hydrodynamics and the Performance of Sailing Yachts". Adlard Coles Nautical, 2009 (cit. on pp. 3, 8, 9).

[2] P. van Oossanen. "Predicting the speed of sailing yachts". In: *SNAME transactions* 101 (1993), pp. 337–397 (cit. on p. 3).

[3] A. B. Philpott, R. M. Sullivan, and P. S. Jackson. "Yacht velocity prediction using mathematical programming". In: *European Journal of Operational Research* (1993) (cit. on p. 4).

[4] C. Böhm. "A Velocity Prediction Procedure for Sailing Yachts With a Hydrodynamic Model Based on Integrated Fully Coupled Ranse-free-surface Simulations". PhD thesis. 2014 (cit. on pp. 4, 5, 10).

[5] K. S. M. Davidson. *Some Experimental Studies of the Sailing Yacht*. Tech. rep. Scheepsbouwkunde, Lab V, 1936 (cit. on pp. 4, 5).

[6] J. Gerritsma, R. Onnink, and A. Versluis. "Geometry, resistance and stability of the Delft Systematic Yacht hull series". In: *International Shipbuilding Progress*. Vol. 28. 328. Delft University of Technology. Delft, The Netherlands, 1981 (cit. on p. 4).

[7] J. A. Keuning and M. Katgert. "A bare hull resistance prediction method derived from the results of the delft systematic yacht hull series extended to higher speeds". In: *International Conference on Innovation in High Performance Sailing Yachts*. Delft University of Technology, Netherands. Lorient, France, 2008 (cit. on pp. 4, 10, 31, 40, 61, 62).

[8] E. P. Warner. "The aerodynamics of yacht sails". In: *Read at the 33rd general meeting of the Society of Naval Architects and Marine Engineers, SNAME, New York, USA* (1926) (cit. on p. 5).

[9] R. G. Flay. "A twisted flow wind tunnel for testing yacht sails". In: *Journal of Wind Engineering and Industrial Aerodynamics* 63.1-3 (1996), pp. 171–182 (cit. on p. 5).

[10] J. H. Milgram and D. B. Peters. "Modeling IACC Sail Forces by Combining Measurements with CFD". In: *11th Chesapeake Sailing Yacht Symposium*. 1993 (cit. on p. 5).

[11] J. H. Milgram. "Naval architecture technology used in winning the 1992 America's Cup match". In: *SNAME Transactions, Volume 101*. Massachusetts Institute of Technology, MIT, Cambridge, USA, 1993, pp. 399–436 (cit. on p. 5).

[12] J. Milgram. "The Aerodynamic of sails". In: *Proceedings of the 7th Symposium on Naval Hydrodynamic*. 1968, pp. 1397–1434 (cit. on p. 6).

[13] I. M. Viola and R. G. J. Flay. "Pressure Distributions on Sails Investigated Using Three Methods: On-Water Measurements, Wind-Tunnel Measurements, and Computational Fluid Dynamics". In: *20th Chesapeake Sailing Yacht Symposium*. 2011 (cit. on p. 6).

[14] J.-B. R. G. Souppez, J. M. M.-A. Dewavrin, F. Gohier, and G. Borba Labi. "Hydrofoil Configurations for Sailing Superyachts: Hydrodynamics, Stability and Performance". In: *proceedings of Design & Construction of Super and Mega Yachts. The Royal Institution of Naval Architects*. Genoa, Italy, 2019 (cit. on pp. 6, 9–11).

[15] J. Den Ouden and M. Katgert. *A User Manual for the Delft Systematic Yacht Hull Series Database website*. Tech. rep. Delft: Delft University of Technology, 2013 (cit. on p. 6).

[16] A. Guell and J.-B. R. G. Souppez. "Combining Modern Hydrofoils with Wooden Classic". In: *the proceedings of the British Conference of Undergraduate Research*. Sheffield, UK, 2018 (cit. on p. 9).

[17] J.-B. R. G. Souppez. "Development and Validation of a Computational Fluid Dynamics Hydrodynamic Model of the Stewart 34 for Velocity Prediction Program Applications". Master Thesis. The University of Auckland., 2014 (cit. on pp. 9, 10).

[18] W. Froude. *Experiments of the surface-friction experienced by a plane moving through water*. Tech. rep. London: the British Association for the Advancement of Science, 1872 (cit. on p. 9).

[19] O. Reynolds. "An experimental Investigation of the circumstances which determine whether the motion of water shall be direct or sinuous, and of the law of resistance in parallel channels". In: *Philosophical transactions of the Royal Society of London* (1883) (cit. on p. 9).

[20] J. B. Hadler. *Coefficients for International Towing Tank Conference 1957 Model-Ship Correlation Line*. Tech. rep. Washington 7 DC, USA: David Taylor Model Basin, Navy Department, Hydromechanics Laboratory, 1957 (cit. on pp. 10, 61).

[21] *ORC VPP Documentation 2019*. Tech. rep. Offshore Racing Congress, 2019 (cit. on pp. 10, 31, 43, 51).

[22] I. M. Viola. "Downwind sail aerodynamics: A CFD investigation with high grid resolution". In: *Ocean Engineering* 36.12-13 (Sept. 2009), pp. 974–984 (cit. on p. 10).

[23] I. M. Viola, S. Bartesaghi, T. Van-Renterghem, and R. Ponzini. "Detached Eddy Simulation of a sailing yacht". In: *Ocean Engineering* 90 (Nov. 2014), pp. 93–103 (cit. on p. 10).

[24] A. Arredondo-Galeana and I. M. Viola. "The leading-edge vortex of yacht sails". In: *Ocean Engineering* 159 (July 2018), pp. 552–562 (cit. on p. 10).

[25] I. M. Viola and A. Arredondo Galeana. "The leading-edge vortex of yacht sails". English. In: *International Conference on Innovation in High Performance Sailing Yachts*. Lorient, France, 2017 (cit. on p. 10).

[26] G. S. Hazen. "A model of sail aerodynamics for diverse rig types". In: *New England sailing yacht symposium* (1980) (cit. on pp. 11, 43, 75).

[27] C. M. D. Mauleverer. "The development of a novel CFD based surrogate model to refine the ORC upwind aero model". Final year dissertation, Solent University. 2018 (cit. on pp. 11, 12).

[28] I. Focarile and J.-B. R. G. Souppez. "Aerodynamic Investigation of Sailing Yachts". English. In: *British Conference of Undergraduate Research*. Cardiff, United Kingdom, 2018 (cit. on p. 11).

[29] J.-B. R. G. Souppez. "On the Applications of Modern Naval Architecture Techniques to Historical Crafts". In: *RINA - Historic Ships*. 2016 (cit. on p. 11).

[30] J. Thomas and J.-B. R. G. Souppez. "Comparative Performance Prediction of Historical Thames A Rater Class Designs". In: *RINA - Historic Ships*. London, United Kingdom, July 2018 (cit. on p. 11).

[31] S. Harries and C. Abt. "CAESES—The HOLISHIP Platform for Process Integration and Design Optimization: Volume 1: Optimisation of Ship Design and Operation for Life Cycle". In: *A Holistic Approach to Ship Design*. Springer, July 2019, pp. 247–293 (cit. on p. 12).

[32] T. Peart, S. Nava, N. Aubin, J. Cater, and S. Norris. "Multi-fidelity surrogate models for VPP aerodynamic input data". In: *The 5th Int. Conference on Innovation in High Performance Sailing Yachts and Sail-Assisted Ship Propulsion*. Gothenburg, Sweden, 2020, pp. 37–48 (cit. on p. 12).

[33] *Bootstrap: the world's most popular front-end open source toolkit*. URL: `https://getbootstrap.com/docs/4.5/components/dropdowns/` (visited on 07/15/2020) (cit. on p. 15).

[34] OWASP. *The Open Web Application Security Project®*. URL: `https://owasp.org/` (visited on 07/15/2020) (cit. on p. 15).

[35] OWASP. *Password Storage Cheat Sheet*. URL: https://cheatsheetseries.owasp.org/cheatsheets/Password_Storage_Cheat_Sheet.html (visited on 07/15/2020) (cit. on p. 16).

[36] A. S. Association. *Understanding Sailboats and Sailing - the sails*. URL: https://asa.com/news/2019/03/14/understanding-sailboats-the-sails/ (visited on 08/02/2020) (cit. on p. 32).

[37] N. Sail. *Sail Plan Dimensions*. URL: https://www.northsails.com/sailing/en/2020/02/sail-plan-dimensions-north-sails (visited on 07/30/2020) (cit. on p. 33).

[38] P. Carbonnelle. *Popularity of Programming Language*. URL: http://pypl.github.io/PYPL.html (visited on 08/01/2020) (cit. on p. 38).

[39] S. Behnel, R. Bradshaw, C. Citro, L. Dalcin, D. S. Seljebotn, and K. Smith. "Cython: The best of both worlds". In: *Computing in Science and Engineering* 13.2 (Mar. 2011), pp. 31–39 (cit. on p. 38).

[40] S. Behnel, R. W. Bradshaw, and D. S. Seljebotn. "Cython Tutorial". In: *Proceedings of the 8th Python in Science Conference*. 2009 (cit. on p. 39).

[41] R. Kern. *Line-by-line profiling for Python*. URL: https://github.com/rkern/line_profiler (visited on 07/23/2020) (cit. on p. 39).

[42] J. Gerritsma. "Course keeping qualities and motions in waves of a sailing yacht". In: *3rd AIAA Symposium Aero, Hydronautics of Sailing*. Redondo Beach, California, USA, 1971 (cit. on pp. 40, 61).

[43] J.-B. R. G. Souppez. "Design and Production of a Wooden Thames A Rater Class Sailing Yacht". Master thesis, The University of Auckland. July 2015 (cit. on pp. 40, 61).

[44] J.-B. R. G. Souppez. "Sailing equilibrium lecture slides". Southampton Solent University, 2019 (cit. on pp. 45, 75).

[45] J. Keuning and B. Sonnenberg. "Approximation of the hydrodynamic forces on a sailing yacht based on the Delft Systematic Yacht Hull Series". In: *15th International Symposium on Yacht Design and Yacht Construction HISWA*. Amsterdam, 1998, pp. 99–152 (cit. on pp. 61, 62, 64–66).

[46] S. F. Hoerner. "Fluid-Dynamic Drag: Theoretical, Experimental and Statistical Information". Hoerner Fluid Dynamics, 1965 (cit. on p. 63).

[47] J. Keuning and B. Sonnenberg. "Approximation of the Calm Water Resistance on a Sailing Yacht Based on the Delft Systematic Yacht Hull Series". In: *14th Chesapeake Sailing Yacht Symposium*. Annapolis, USA, 1999 (cit. on pp. 63, 65, 66).

[48] S. Wallis. "Marine craft design and development: estimating Keels and Ballast". Ed. by Southampton Solent University. Southampton, 2013 (cit. on p. 63).

[49] J. Keuning and B. Verwerft. “A new method for prediction of the side force on keel and rudder of a sailing yacht based on the results of the Delft systematic yacht hull series”. In: *19th Chesapeake Sailing Yacht Symposium*. Annapolis, Maryland, USA, 2009 (cit. on p. 67).

Appendix A

DSYHS Calculations

The calculations used to validate the hull resistance of the VPP are given below. The hydrodynamic comparison model is built on the Delft Systematic Yacht Hull Series as originally developed in the 1970s by Gerritsma [42] and continued by Keuning et al. [7]; the various elements have been compiled by Souppez [43].

A.1 Upright Bare Hull

The hull friction Rfh developed in the DSYHS is given by [45]:

$$Rfh = \frac{1}{2} \cdot \rho \cdot V^2 \cdot Sc \cdot Cf \,. \tag{A.1}$$

The friction coefficient used in this equation is based on the ITTC 1957 formulae [20] with a correction factor of 0.7 applied on the water line length in the Reynolds number. Modern hulls such as the TP52 can take a larger correction of 0.9 here we stick to 0.7.

$$Cf = \frac{0.0075}{(\log_{10} Rn - 2)^2} \,, \tag{A.2}$$

$$Rn = \frac{\rho \cdot V \cdot (0.7 \cdot Lwl)}{\mu} \,. \tag{A.3}$$

The reason behind this correction lies in the fact that the ITTC 57 coefficients are built for ships, which traditionally have a parallel midbody that yachts do not have. The reduced length decreases the Reynolds number thus increasing the friction coefficient.

The wetted surface area Sc is approximated as [45]:

$$Sc = \left(1.97 + 0.171 \cdot \frac{Bwl}{Tc}\right) \cdot \left(\frac{0.65}{Cm}\right)^{(1/3)} \cdot (\nabla_C \cdot Lwl)^{(1/2)} \,. \tag{A.4}$$

The residuary Resistance Rrh is defined in [45] was updated in [7]. The original equation was not suitable the recent hulls added to the series when sailing at high Froude numbers. The bare hull residuary resistance is then defined as [7]:

$$\begin{aligned} \frac{\mathrm{Rrh}}{\nabla c \cdot \rho \cdot g} =& a_0 + \left(a_1 \cdot \frac{LCB_{fpp}}{Lwl} + a_2 \cdot Cp + a_3 \cdot \frac{\nabla c^{2/3}}{Aw} + a_4 \cdot \frac{Bwl}{Lwl} \right. \\ & \left. + a_5 \cdot \frac{LCB_{fpp}}{LCF_{fpp}} + a_6 \cdot \frac{Bwl}{Tc} + a_7 \cdot Cm\right) \cdot \frac{\nabla c^{1/3}}{Lwl} \,. \end{aligned} \tag{A.5}$$

Where the a_i are regression coefficients given in table A.1.

Table A.1: DSYHS upright hull residuary resistance regression coefficients [7]

Fn	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7
0.15	-0.0005	0.0023	-0.0086	-0.0015	0.0061	0.0010	0.0001	0.0052
0.20	-0.0003	0.0059	-0.0064	0.0070	0.0014	0.0013	0.0005	-0.0020
0.25	-0.0002	-0.0156	0.0031	-0.0021	-0.0070	0.0148	0.0010	-0.0043
0.30	-0.0009	0.0016	0.0337	-0.0285	-0.0367	0.0218	0.0015	-0.0172
0.35	-0.0026	-0.0567	0.0446	-0.1091	-0.0707	0.0914	0.0021	-0.0078
0.40	-0.0064	-0.4034	-0.1250	0.0273	-0.1341	0.3578	0.0045	0.1115
0.45	-0.0218	-0.5261	-0.2945	0.2485	-0.2428	0.6293	0.0081	0.2086
0.50	-0.0388	-0.5986	-0.3038	0.6033	-0.0430	0.8332	0.0106	0.1336
0.55	-0.0347	-0.4764	-0.2361	0.8726	0.4219	0.8990	0.0096	-0.2272
0.60	-0.0361	0.0037	-0.2960	0.9661	0.6123	0.7534	0.0100	-0.3352

The total upright hull resistance Rth is the sum of the frictional and residuary resistance.

$$Rth = Rfh + Rrh \,. \tag{A.6}$$

A.2 Upright Appendages.

The frictional resistance for the appendages is found with the previously established equation A.1. The Reynolds number can be calculated using the mean chord length of the foil. The viscous resistance is given by [45]:

$$R\nu_{app} = Rf \cdot (1 + k) \,. \tag{A.7}$$

With the form factor for the foil calculated based on Hoerner formula [46]:

$$1 + k = 1 + 2 \cdot \frac{t}{c} + 60 \cdot \left(\frac{t}{c}\right)^4 \tag{A.8}$$

The residuary resistance is calculated only for the keel, it can be neglected for the rudder, the equation is given by [47]:

$$\frac{Rr_{keel}}{\nabla_{keel} \cdot \rho \cdot g} = A_0 + A_1 \cdot \frac{T}{Bwl} + A_2 \cdot \frac{Tc + Zcb_{keel}}{\nabla_{keel}^{1/3}} + A_3 \cdot \frac{\nabla c}{\nabla_{keel}} \,. \tag{A.9}$$

Where A_i are the regression coefficients presented in Table A.2.

Table A.2: DSYHS upright appendages residuary resistance regression coefficients

Fn	A_0	A_1	A_2	A_3
0.20	-0.00104	0.00172	0.00117	-0.00008
0.25	-0.00550	0.00597	0.00390	-0.00009
0.30	-0.01110	0.01421	0.00069	0.00021
0.35	-0.00713	0.02632	-0.00232	0.00039
0.40	-0.03581	0.08649	0.00999	0.00017
0.45	-0.00470	0.11592	-0.00064	0.00035
0.50	0.00553	0.07371	0.05991	-0.00114
0.55	0.04822	0.00660	0.07048	-0.00035
0.60	0.01021	0.14173	0.06409	-0.00192
Coefficients are multiplied by 1000				

The volume of the keel ∇_{keel} and its vertical centre of buoyancy Zcb_{keel} required in the residuary resistance of the keel those can be estimated according to [48]:

$$\nabla_{keel} = SAC \cdot \frac{t}{c} \cdot \frac{b}{3} \cdot \left(Cr^2 + Cr \cdot Ct + Ct^2\right) \tag{A.10}$$

$$Zcb_{keel} = \frac{b}{4} \cdot \left(\frac{1 + 2 \cdot Tr + 3 \cdot Tr^2}{1 + Tr + Tr^2}\right) \tag{A.11}$$

The total upright resistance Rt accounting the hull and appendages is calculated with the following sum:

$$Rt = Rth + R\nu_{keel} + Rr_{keel} + R\nu_{rudder} \tag{A.12}$$

The upright resistance is not enough to model the yacht in action on the water. Indeed while sailing yachts heel in response to the wind blowing in the sails. An additional term to the resistance is provided in the method to account for this phenomenon.

A.3 Heeled Bare Hull

The DSYHS assumes the change in frictional resistance is due to the new wetted surface ares $Sc\varphi$ defined in [45] as:

$$Sc\varphi = Sc_{(\varphi=0°)} \cdot \left(1 + \frac{1}{100}\left(s_0 + s_1 \cdot \frac{Bwl}{Tc} + s_2 \cdot \left(\frac{Bwl}{Tc}\right)^2 + s_3 \cdot Cm\right)\right) . \quad \text{(A.13)}$$

The regression coefficients s_i are given table A.3 for angles ranging from 5° to 35° in 5° increments. For angles in between the values given in the table the wetted surface area calculated with a linear interpolation of the coefficients between the two existing heeling angles.

Table A.3: DSYHS heeled wetted surface area regression coefficients [45]

φ	s_0	s_1	s_2	s_3
5°	-4.112	0.054	-0.027	6.329
10°	-4.522	-0.132	-0.077	8.738
15°	-3.291	-0.389	-0.118	8.949
20°	1.850	-1.200	-0.109	5.364
25°	6.510	-2.305	-0.066	3.443
30°	12.334	-3.911	0.024	1.767
35°	14.648	-5.182	0.102	3.497

The friction resistance of the heeled bare hull Rfh_φ is computed by injecting $Sc\varphi$ in the friction resistance equation A.1.

For the residuary resistance of the hull, the correction term is based on a new regression formula. All the hulls in the series have been tested with a heel angle of 20°. These tests have allowed to establish residuary delta $\Delta Rrh_{(\varphi=20°)}$ between 20° of heel and the upright condition:

$$\frac{\Delta Rrh_{(\varphi=20^\circ)}}{\nabla c \cdot \rho \cdot g} = u_0 + u_1 \cdot \frac{Lwl}{Bwl} + u_2 \cdot \frac{Bwl}{Tc} + u_3 \cdot \left(\frac{Bwl}{Tc}\right)^2 + u_4 \cdot LCB + u_5 \cdot LCB^2 \,. \quad \text{(A.14)}$$

Where u_i are the regression coefficients given in the table A.4

Table A.4: DSYHS 20° of heel hull delta residuary resistance regression coefficients [45].

Fn	u_0	u_1	u_2	u_3	u_4	u_5
0.25	-0.0268	-0.0014	-0.0057	0.0016	-0.0070	-0.0017
0.30	0.6628	-0.0632	-0.0699	0.0069	0.0459	-0.0004
0.35	1.6433	-0.2144	-0.1640	0.0199	-0.0540	-0.0268
0.40	-0.8659	-0.0354	0.2226	0.0188	-0.5800	-0.1133
0.45	-3.2715	0.1372	0.5547	0.0268	-1.0064	-0.2026
0.50	-0.1976	-0.1480	-0.6593	0.1862	-0.7489	-0.1648
0.55	1.5873	-0.3749	-0.7105	0.2146	-0.4818	-0.1174
Coefficients multiplied by 1000						

The residuary delta for any heel angle φ up to 30° can be calculated with the following equation:

$$\Delta \,\mathrm{Rrh}_{\varphi} = \Delta \,\mathrm{Rrh}_{(\varphi=20^\circ)} \cdot 6 \cdot \varphi^{1.7} \quad \text{(A.15)}$$

note the angle φ in this equation needs to be in radians.

Finally, the total heeled resistance of the hull is given by:

$$Rth\varphi = Rfh_{\varphi} + Rrh + \Delta Rrh_{\varphi} \quad \text{(A.16)}$$

A.4 Heeled Appendages

Heeled appendages are corrected with a similar delta on the residuary resistance [47] :

$$\frac{\Delta Rr\varphi k}{\nabla_{keel} \cdot \rho \cdot g} = \left(H_1 \cdot \frac{Tc}{T} + H_2 \cdot \frac{Bwl}{Tc} + H_3 \cdot \frac{Tc}{T} \cdot \frac{Bwl}{Tc} + H_4 \cdot \frac{Lwl}{{\nabla_{keel}}^{1/3}}\right) \cdot Fn^2 \cdot \varphi \,. \quad \text{(A.17)}$$

The regression coefficients used in this case are given in the table A.5.

Table A.5: DSYHS heeled appendages delta residuary resistance regression coefficients [47]

H_1	H_2	H_3	H_4
-3.5837	-0.0518	0.5958	0.2055

The total resistance $Rt\varphi$ considering the hull and appendages in heeled conditions is thus:

$$Rt\varphi = Rth\varphi + Rv_{keel} + Rr\varphi_{keel} + Rv_{rudder} \, . \tag{A.18}$$

One last additional term is considered in the resistance calculation it concerns yawed appendages also known as leeway where foils are angled to counteract the yaw moment generated by the sails.

A.5 Yawed Appendages

The appendages generate lift to counteract the yaw rotation of the yacht, doing so it induces a drag component and a side force. Both can be evaluated with the following method.

Starting with the drag the effective draught T_e of the keel is evaluated according to [45]:

$$\frac{Te}{T} = \left(A_1 \cdot \frac{Tc}{T} + A_2 \cdot \left(\frac{Tc}{T} \right)^2 + A_3 \cdot \frac{Bwl}{Tc} + A_4 \cdot TR \right) \cdot (B_0 + B_1 \cdot Fn) \, . \tag{A.19}$$

With the coefficients A_i and B_i given in the table A.6

Table A.6: DSYHS effective draught regression coefficients

φ	A_1	A_2	A_3	A_4	B_1	B_2
0°	3.7455	-3.6246	0.0589	-0.0296	1.2306	-0.7256
10°	4.4892	-4.8454	0.0294	-0.0176	1.4231	-1.2971
20°	3.9592	-3.9804	0.0283	-0.0075	1.5450	-1.5622
30°	3.4891	-2.9577	0.0250	-0.0272	1.4744	-1.3499

The effective aspect ratio is determined as [45]:

$$AR_e = \frac{T_e^2}{A_{lat}} \, , \tag{A.20}$$

where A_{lat} is the lateral area of the keel defined as the product of the mean chord $\bar{c}$ and the draught T

The lift and drag coefficients are then calculated with the following equations:

$$C_L = \frac{2\pi \cdot \lambda}{1 + 3/AR_e} \tag{A.21}$$

$$C_{Di} = \frac{C_L^2}{\pi \cdot AR_e} \tag{A.22}$$

With these coefficients the side force Fh and drag Ri are simply derived from :

$$Fh = C_L \cdot \frac{1}{2} \cdot \rho \cdot V^2 \cdot A_{lat} \tag{A.23}$$

$$Ri = C_{Di} \cdot \frac{1}{2} \cdot \rho \cdot V^2 \cdot A_{lat} \tag{A.24}$$

The induced drag is added to the viscous and the residuary resistance to yield the total resistance of the keel:

$$Rt_{keel} = Rv_{keel} + Rr\varphi_{keel} + Ri_{keel}\,. \tag{A.25}$$

The same approach is pursued for the rudder, however, small correcting factors must be applied as the rudder operates in the downwash of the keel [49]. The flow velocity seen by the rudder is reduced to 90% of the keel flow velocity (i.e. boat speed) and the leeway angle at the rudder is taken as half the value seen by the keel.

The resulting hull resistance is given on the fig 4.7 while the python script for this model is given in the listing A.1

Listing A.1: Python script building the hydrodynamic forces of the yacht.

```
# -*- coding: utf-8 -*-
"""
Created on Tue Jun 30 16:04:44 2020

@author: PierreCourtel
"""
import numpy as np
import pandas as pd

import matplotlib.pyplot as plt
from matplotlib import rc
rc('font',**{'family':'sans-serif','sans-serif':['Helvetica']})
## for Palatino and other serif fonts use:
#rc('font',**{'family':'serif','serif':['Palatino']})
rc('text', usetex=True)

def main():

    plt.close('all')

    dctControl = {}
    dctControl['Plot_xAxis'] = 'BoatSpeed' #Froude
```

```
dctConstants ={}
dctConstants['rho_water'] = 1025
dctConstants['gravity']   = 9.81
dctConstants['kin_Visc']  = 1.002e-6
dctConstants['dyn_Visc']  = 1.002e-3


dctBoatData = {}
dctBoatData['WaterLineLength'] = 12.2864
dctBoatData['WaterLineBeam']   =  2.8060
dctBoatData['DraftCanoe']      =  0.5080
dctBoatData['DraftYacht']      =  2.4124
dctBoatData['DispCanoe']       =  6.8969
dctBoatData['MaxSectArea']     =  1.0623
dctBoatData['PrismCoeff']      =  0.5309
dctBoatData['WaterPlaneArea']  = 21.8944
dctBoatData['LeewayAngleDeg']  =  3.5000
dctBoatData['LongCentBuoy']    = -3.0900e-2
dctBoatData['LongCentFlot']    = -7.1700e-2
dctBoatData['LongCentBuoyfpp'] =  0.5487*dctBoatData['WaterLineLength']
dctBoatData['LongCentFlotfpp'] =  0.5717*dctBoatData['WaterLineLength']

#Rudder
dctBoatData['RudWettedArea']   =  0.7303
dctBoatData['RudChord']        =  0.3414
dctBoatData['RudThick']        =  0.1562*dctBoatData['RudChord']

#Keel
dctBoatData['KeelChordTip']    =  1.2868
dctBoatData['KeelChordRoot']   =  0.9216
dctBoatData['KeelChordAvg']    =  1.0815
dctBoatData['KeelThick']       =  0.1321*dctBoatData['KeelChordAvg']
dctBoatData['KeelWettedArea']  =  2.1491
dctBoatData['KeelDisp']        = dctBoatData['KeelWettedArea'] * dctBoatData['KeelThick']
dctBoatData['KeelCentBuyo']    = dctBoatData['DraftCanoe'] + (dctBoatData['DraftYacht']-dctBoatData['DraftCanoe'])/2

#Bulb
fltBulbRadius = 0.2
fltBulbLength = 1.5
dctBoatData['BulbDisp']        = np.pi*np.power(fltBulbRadius,2)*fltBulbLength*0.6
dctBoatData['BulbThick']       = 0.35 #number guessed based on Labi 2019
dctBoatData['BulbChord']       = 2.4  #number guessed based on Labi 2019
dctBoatData['BulbWettedAreaUpright'] = 2*np.pi*fltBulbRadius*fltBulbLength*0.85

dctBoatData['MidAreaCoeff']    = dctBoatData['MaxSectArea']/(dctBoatData['WaterLineBeam']*dctBoatData['DraftCanoe'])
dctBoatData['HullWettedAreaUpright'] = WettedAreaUpright(dctBoatData)

dfPolyHeel    = pd.read_csv('HeelPolyCoeff.csv')
dfPolyUpright = pd.read_csv('UprightPolyCoeff.csv')[dfPolyHeel.columns]
dfPolyKeel    = pd.read_csv('KeelPolyCoeff.csv')[dfPolyHeel.columns]
dfPolySideF   = pd.read_csv('SideForcePolyCoeff.csv')
dfPolyWetArea = pd.read_csv('WettedSurfacePolyCoeff.csv')
dfPolyEff     = pd.read_csv('EffectivePolyCoeff.csv')


arrPolyUpright   = dfPolyUpright.to_numpy()[:,1:]
arrPolyHeel      = dfPolyHeel.to_numpy()[:,1:]/1000 #
arrPolyKeel      = dfPolyKeel.to_numpy()[:,1:]/1000 #
arrPolySideF     = dfPolySideF.to_numpy()[:,1:]
arrPolyWetArea   = dfPolyWetArea.to_numpy()[:,1:]
arrPolyEff       = dfPolyEff.to_numpy()[:,1:]

lstFroudes   = dfPolyHeel.columns.to_list()[1:]
lstFroudes   = [float(i) for i in lstFroudes] #casting the strings to floats
arrFroudes   = np.array(lstFroudes)
arrBoatspeed = np.array([])
arrReynolds  = np.array([])
arrFricCoef  = np.array([])
arrRrh       = np.array([])

arrBoatspeed = arrFroudes * np.sqrt(dctBoatData['WaterLineLength'] * dctConstants['gravity'])
arrReynolds  = 0.7 * arrBoatspeed * dctBoatData['WaterLineLength'] / dctConstants['kin_Visc']
arrFricCoef  = 0.075/np.power((np.log10(arrReynolds)-2),2)


### Hull Resistance

# Upright Friction Resistance
```

```
arrRfh = 0.5*dctConstants['rho_water']*np.power(arrBoatspeed,2)*dctBoatData['HullWettedAreaUpright']*arrFricCoef

# Upright Residuary Resistance
arrRrh = residuaryResistance(dctBoatData, dctConstants, arrPolyUpright)

arrRth = arrRfh + arrRrh



### Rudder Resistance
fltRudChord      = dctBoatData['RudChord']
fltRudThick      = dctBoatData['RudThick']
fltLeewayRud     = dctBoatData['LeewayAngleDeg']*0.5
arrRf_Rud        = 0.5*dctConstants['rho_water']*np.power(arrBoatspeed,2)*dctBoatData['RudWettedArea']*arrFricCoef
arrOnePlusK_Rud  = 1 + 2 * fltRudThick/fltRudChord + 60 * np.power(fltRudThick/fltRudChord,4)
arrRv_Rud        = arrRf_Rud * arrOnePlusK_Rud
arrRi_Rud        = InducedResistance(fltLeewayRud,dctBoatData, dctConstants, 0.9*arrBoatspeed, arrPolyEff, arrFroudes)
arrRt_Rud        = arrRv_Rud + arrRi_Rud

### Keel Resistance
fltKeelChord     = dctBoatData['KeelChordAvg']
fltKeelThick     = dctBoatData['KeelThick']
fltLeeway        = dctBoatData['LeewayAngleDeg']
arrRf_Keel       = 0.5 * dctConstants['rho_water']*np.power(arrBoatspeed,2)*dctBoatData['KeelWettedArea']*arrFricCoef
arrOnePlusK_Keel = 1 + 2 * fltKeelThick/fltKeelChord + 60 * np.power(fltKeelThick/fltKeelChord,4)
arrRv_Keel       = arrRf_Keel * arrOnePlusK_Keel
arrRr_Keel       = residuaryResistanceKeel(dctBoatData, dctConstants, arrPolyKeel)
arrRi_Keel       = InducedResistance(fltLeeway,dctBoatData, dctConstants, arrBoatspeed, arrPolyEff, arrFroudes)
arrRt_Keel       = arrRv_Keel + arrRr_Keel + arrRi_Keel



phi_heeldeg = np.array([0, 10, 20 , 30])
phi_heels   = phi_heeldeg*np.pi/180
lstMarkers  = ['o','s','d','*']
fig2, ax2   = plt.subplots()

if dctControl['Plot_xAxis'] == 'BoatSpeed':
    arrXaxis  = arrBoatspeed*1.94384
    strXlabel = 'Boat Speed [kts]'
elif dctControl['Plot_xAxis'] == 'Froude':
    arrXaxis  = arrFroudes
    strXlabel = 'Froude nbr'

### Change in residuary resistance with Heel
for intPhiIx, phi_heel in enumerate(phi_heels):


    arrDeltaRrh_phi = HeelResistance(phi_heel, dctBoatData, dctConstants, arrPolyHeel)
    arrDeltaRr_Keel = KeelHeelChange(phi_heel, dctBoatData, dctConstants, arrFroudes)

    # beta = dctBoatData['LeewayAngleDeg']*np.pi/180
    # Fh   = SideForceProduction(beta, 10*intPhiIx, dctBoatData, dctConstants, arrBoatspeed, arrPolyWetArea, arrPolySideF)

    Total_Resistance = arrRth + arrDeltaRrh_phi + arrRt_Keel + arrDeltaRr_Keel + arrRt_Rud


    ax2.plot(arrXaxis,
            Total_Resistance,
            marker = lstMarkers[intPhiIx],
            label=r"$\varphi$" ' ='+str(phi_heeldeg[intPhiIx])+'°')

    # fig, ax = plt.subplots()
    # ax.plot(arrXaxis, Total_Resistance          , marker='o', label='Heeled with appendages')
    # ax.plot(arrXaxis, arrRth + arrDeltaRrh_phi , marker='s', label='Heeled , no appendages')
    # ax.plot(arrXaxis, arrRth                    , marker='d', label='Upright, no appendages')

    # ax.set(xlabel=strXlabel, ylabel='Resistance [N]',
    #     title='Total Resistance, Heel :'+str(round(phi_heel*180/np.pi)) +'° with appendages(rudder, keel, bulb)')
    # ax.grid()
    # ax.legend(loc='upper left')
    # plt.show()


ax2.set(xlabel=strXlabel        ,
        ylabel='Resistance [N]' ,
        title='DSYHS total resistance with appendages(rudder, keel, bulb)')
ax2.grid()
```

```
    handles, labels = ax2.get_legend_handles_labels()
    ax2.legend(handles[::-1], labels[::-1], title='Heeling Angles', loc='upper left')
    plt.savefig('test.eps', format='eps')
    plt.show()


def SideForceProduction(beta, phi, dctBoatData, dctConstants, arrBoatspeed, arrPolyWetArea, arrPolySideF):

    T  = dctBoatData['DraftYacht']
    Tc = dctBoatData['DraftCanoe']


    rho_water = dctConstants['rho_water']


    if phi == 0:
        Sc = WettedAreaUpright(dctBoatData)
        arrPolySideF = arrPolySideF[:,0]
    elif phi in [10, 20 , 30]:
        intIndex = [10, 20 , 30].index(phi)
        arrPolyWetArea = arrPolyWetArea[:,intIndex]
        arrPolySideF = arrPolySideF[:,intIndex]
        Sc = WettedAreaHeel(dctBoatData, arrPolyWetArea)
    else:
        return None


    b1 = arrPolySideF[0]
    b2 = arrPolySideF[1]
    b3 = arrPolySideF[2]
    b4 = arrPolySideF[3]

    Fh = beta * 0.5 * rho_water * np.power(arrBoatspeed,2) * Sc / np.cos(phi*np.pi/180)

    Fh *= (
             b1 * np.power(T,2) /Sc             \
           + b2 * np.power(T,4) /np.power(Sc,2) \
           + b3 * Tc/T                          \
           + b4 * Tc*(T/Sc)                     \
           )
    print('Phi: '+str(phi))
    print('WettedArea :',Sc)
    print('Side Force :', Fh)
    print('\n ')
    return Fh

def residuaryResistanceKeel(dctBoatData, dctConstants, arrPolyKeel):
    Disp_Keel = dctBoatData['KeelDisp']
    T         = dctBoatData['DraftYacht']
    Bwl       = dctBoatData['WaterLineBeam']
    Tc        = dctBoatData['DraftCanoe']
    Zcbk      = dctBoatData['KeelCentBuyo']
    Disp_C    = dctBoatData['DispCanoe']

    rho_water = dctConstants['rho_water']
    gravity   = dctConstants['gravity']


    A0 = arrPolyKeel[0]
    A1 = arrPolyKeel[1]
    A2 = arrPolyKeel[2]
    A3 = arrPolyKeel[3]

    Rrkeel = Disp_Keel * rho_water * gravity * (
                                            A0                                  \
                                          + A1 * T/Bwl                          \
                                          + A2 * (Tc + Zcbk)/np.power(Disp_Keel,3) \
                                          + A3 * Disp_C/Disp_Keel               \
                                          )
    return Rrkeel


def InducedResistance(fltLeeway,dctBoatData, dctConstants, arrBoatspeed, arrPolyEff, arrFroudes):

    rho_water = dctConstants['rho_water']

    T   = dctBoatData['DraftYacht']
    Tc  = dctBoatData['DraftCanoe']
    Bwl = dctBoatData['WaterLineBeam']
```

```
    TR  = dctBoatData['KeelChordTip']/dctBoatData['KeelChordRoot']


    #Interpolation for the correct Leeway
    arrYawedPolyEff = LinearInterpolation(fltLeeway, arrPolyEff)

    A1 = arrYawedPolyEff[0]
    A2 = arrYawedPolyEff[1]
    A3 = arrYawedPolyEff[2]
    A4 = arrYawedPolyEff[3]
    B0 = arrYawedPolyEff[4]
    B1 = arrYawedPolyEff[5]


    Te = T * (A1 * (Tc/T) + A2 * np.power(Tc/T,2) + A3 * (Bwl/Tc) + A4 * TR)*(B0 + B1*arrFroudes)

    Alat = dctBoatData['KeelChordAvg'] * T

    ARe = np.power(Te,2)/Alat

    fltLeewayRad = fltLeeway*np.pi/180
    Cl = 2*np.pi*fltLeewayRad/(1+2/ARe)

    Cdi = np.power(Cl,2)/(np.pi * ARe)

    Ri = 0.5 * Cdi * rho_water * np.power(arrBoatspeed,2) * Alat

    return Ri

def LinearInterpolation(fltLeeway, arrPolyEff):

    intIx = int(fltLeeway/10) #The leeway angle is between intIx and intIx + 1

    #Interpolate the regression coefficients for the Leeway angle.
    intLeeway0 = 10*intIx
    intLeeway1 = 10*(intIx+1)

    arrRegCoeff0 = arrPolyEff[intIx]
    arrRegCoeff1 = arrPolyEff[intIx+1]

    x  = fltLeeway
    x0 = intLeeway0
    x1 = intLeeway1
    y0 = arrRegCoeff0
    y1 = arrRegCoeff1

    y = y0 + (x-x0)*(y1-y0)/(x1-x0)

    return y

def residuaryResistance(dctBoatData, dctConstants, arrPolyUpright):
    Lwl    = dctBoatData['WaterLineLength']
    Bwl    = dctBoatData['WaterLineBeam']
    Tc     = dctBoatData['DraftCanoe']
    LCB    = dctBoatData['LongCentBuoyfpp']
    LCF    = dctBoatData['LongCentFlotfpp']
    Cp     = dctBoatData['PrismCoeff']
    Disp_C = dctBoatData['DispCanoe']
    Aw     = dctBoatData['WaterPlaneArea']
    Cm     = dctBoatData['MidAreaCoeff']

    rho_water = dctConstants['rho_water']
    gravity   = dctConstants['gravity']

    a0 = arrPolyUpright[0]
    a1 = arrPolyUpright[1]
    a2 = arrPolyUpright[2]
    a3 = arrPolyUpright[3]
    a4 = arrPolyUpright[4]
    a5 = arrPolyUpright[5]
    a6 = arrPolyUpright[6]
    a7 = arrPolyUpright[7]

    Rrh = Disp_C*rho_water*gravity*(
                                a0 + (  a1 * LCB/Lwl                  \
                                      + a2 * Cp                       \
                                      + a3 * np.power(Disp_C, 2/3)/Aw \
                                      + a4 * Bwl/Lwl                  \
                                      + a5 * LCB/LCF                  \
```

```
                                    + a6 * Bwl/Tc                    \
                                    + a7 * Cm                        \
                                    )* np.power(Disp_C, 1/3)/Aw      \
                                )
    return Rrh


###Change in Resistance Due to Heel
def HeelResistance(phi, dctBoatData, dctConstants, arrPolyHeel):
    Lwl     = dctBoatData['WaterLineLength']
    Bwl     = dctBoatData['WaterLineBeam']
    Tc      = dctBoatData['DraftCanoe']
    LCB     = dctBoatData['LongCentBuoy']
    Disp_C = dctBoatData['DispCanoe']

    rho_water = dctConstants['rho_water']
    gravity   = dctConstants['gravity']

    u0 = arrPolyHeel[0]
    u1 = arrPolyHeel[1]
    u2 = arrPolyHeel[2]
    u3 = arrPolyHeel[3]
    u4 = arrPolyHeel[4]
    u5 = arrPolyHeel[5]

    Delta_Rrh_phi20 = Disp_C * rho_water * gravity * (
                                                u0                          \
                                              + u1 * Lwl/Bwl                \
                                              + u2 * Bwl/Tc                 \
                                              + u3 * np.power((Bwl/Tc),2)   \
                                              + u4 * LCB                    \
                                              + u5 * np.power(LCB,2)        \
                                                )
    Delta_Rrh_phi = Delta_Rrh_phi20 * 6 * np.power(phi,1.7)
    return Delta_Rrh_phi

### Change in resistance due to heel for the keel
def KeelHeelChange(phi, dctBoatData, dctConstants, arrFroudes):
    Lwl       = dctBoatData['WaterLineLength']
    Bwl       = dctBoatData['WaterLineBeam']
    Tc        = dctBoatData['DraftCanoe']
    T         = dctBoatData['DraftYacht']
    Disp_Keel = dctBoatData['KeelDisp']

    fltCh = -3.5837 * Tc/T + -0.0518 * Bwl/Tc + 0.5958* Bwl/T + 0.2055* Lwl/np.power(Disp_Keel,1/3)

    rho_water = dctConstants['rho_water']
    gravity   = dctConstants['gravity']

    Delta_Rrk_phi = fltCh * np.power(arrFroudes,2) * phi * Disp_Keel * rho_water * gravity
    return Delta_Rrk_phi

def WettedAreaHeel(dctBoatData, arrPolyWetArea):

    s0 = arrPolyWetArea[0]
    s1 = arrPolyWetArea[1]
    s2 = arrPolyWetArea[2]
    s3 = arrPolyWetArea[3]

    Bwl = dctBoatData['WaterLineBeam']
    Tc  = dctBoatData['DraftCanoe']
    Cm  = dctBoatData['MidAreaCoeff']

    Sc_phi0 = WettedAreaUpright(dctBoatData)

    Sc_phi = Sc_phi0 * (1 +     \
                        1/100 * (
                                    s0                          \
                                  + s1 * Bwl/Tc                 \
                                  + s2 * np.power(Bwl/Tc, 2)    \
                                  + s3 * Cm                     \
                                  )
                        )

    return Sc_phi


def WettedAreaUpright(dctBoatData):
    Lwl     = dctBoatData['WaterLineLength']
```

```
    Bwl     = dctBoatData['WaterLineBeam']
    Tc      = dctBoatData['DraftCanoe']
    Cm      = dctBoatData['MidAreaCoeff']
    Disp_C = dctBoatData['DispCanoe']

    Sc = (1.97+0.171*Bwl/Tc)*np.power(0.65/Cm,1/3)*np.sqrt(Disp_C* Lwl)
    return Sc




#########################################################################################################
#########################################################################################################
# Main Start Point
#########################################################################################################
if __name__ == '__main__':
    main()
```

Appendix B

Aerodynamic ORC force calculations

The python script given in the listing below implements a semi-empirical method developed by Hazen [26] using the ORC lift and drag coefficients as presented by Souppez [44].

Listing B.1: Determination of the drive and side forces using the ORC

```
# -*- coding: utf-8 -*-
"""
Created on Sun Aug  2 17:07:52 2020

@author: PierreCourtel
"""
import numpy as np

def main():
    pAir = 1.23
    kts2ms = 0.5144

    BOA = 2.8060
    FA  = 1.26

    P = 15.025
    E =  5.880
    I = 15.115
    J =  4.840
    BAS = 1.585

    HHB = 0.07
    HUW = 0.84
    HTW = 1.56
    HHW = 2.69
    HQW = 3.81
    HLP = 5.01
    HLU = 15.16

    AreaHead = 0.1125 * HLU * (1.445 * HLP + 2 * HQW+ 2 * HHW + 1.5 * HTW + HUW + 0.5 * HHB)

    MHB  = 1.57
    MUW  = 1.98
    MTW  = 2.71
    MHW  = 3.88
    MQW  = 4.85
    MDL1 = 0.198

    MHWH = P/2 + (MHW - E/2)*E/P
    MQWH = MHWH/2 + (MQW - (E+ MHW)/2)/MHWH * (E - MHW)
    MTWH = (MHWH+P)/2 + (MTW - MHW/2)/(P - MHWH) * MHW
    MUWH = (MTWH+P)/2 + (MUW - MTW/2)/(P - MTWH) * MTW

    AreaMain = (
                   (MQW +  E )/2 *      MQWH       \
                 + (MQW + MHW)/2 *  (MHWH - MQWH)  \
                 + (MHW + MTW)/2 *  (MTWH - MHWH)  \
                 + (MTW + MUW)/2 *  (MUWH - MTWH)  \
                 + (MUW + MHB)/2 *  (  P   - MUWH)  \
                 )

    AreaSailPlan = AreaMain + AreaHead

    ### Hazen Calculation
```

```
    AN_main = 0.5*P*E
    AN_head = 0.5*I*J
    A_head  = 0.5*np.sqrt(np.power(I,2)+np.power(J,2))*HLP


    EHM = P + BAS
    AR = (1.1 * np.power(EHM+FA,2))/(AN_main+AN_head)

    # Taken from the
    AWA    = 17.95
    AWS    = 16.16*kts2ms
    AWArad = AWA*np.pi/180

    CDrag_main = 0.02
    CLift_main = 1.32
    CDrag_head = 0.02
    CLift_head = 1.45


    CL      = (CLift_main * AN_main + CLift_head * AN_head) / (AN_main+AN_head)
    CD_para = (CDrag_main * AN_main + CDrag_head * AN_head) / (AN_main+AN_head)

    CD_indu = 0.02+np.power(CL,2)*(1/(np.pi*AR)+0.005)
    CD_rig  = 1.13 * ((BOA * FA) + (EHM * MDL1))/(AN_main+AN_head)

    CD = CD_indu


    Lift = 0.5 * pAir * (AN_main+AN_head) * np.power(AWS,2) * CL
    Drag = 0.5 * pAir * (AN_main+AN_head) * np.power(AWS,2) * CD

    FD  = Lift*np.sin(AWArad) - Drag*np.cos(AWArad)
    SSF = Lift*np.cos(AWArad) + Drag*np.sin(AWArad)

    print('FD: \n')
    print(FD)
    print('\n SSf: \n')
    print(SSF)
    return


########################################################################################################
########################################################################################################
# Main Start Point
########################################################################################################
if __name__ == '__main__':
    main()
```