Time Optimization for Laser Sailing Races

Gabriela Pardo Arteaga

Technische Universiteit Delft

TUDelft Delft University of Technology

Challenge the future

Time Optimization for Laser Sailing Races

by

Gabriela Pardo Arteaga

in partial fulfillment of the requirements for the degree of

Master of Science
in Bio-Mechanical Design

at the Delft University of Technology,
to be defended publicly on May 28, 2019.

Supervisor:	Prof. dr. ir. A.W. Heemink	TU Delf
Thesis committee:	Prof. dr. C. F. Frans C.T. van der Helm,	TU Delft
	Dr. E. L. Arend L. Schwab,	TU Delft
	Ir. M. Arjen Jansen	TU Delft

Curiosity is the startling
unexpectedness of all beginnings.

Hannah Arendt

SUMMARY

This study develops an algorithm on *MATLAB*® to model the path for sailing competitions and optimize its time-path shaped by wind of sailing races for Lasers (Olympic class). Furthermore, to identify the wind effects with different time-steps sizes on the resulting time-paths and trajectories. For the validation of the algorithm, this research compares the results between the algorithm using alternative scenarios with the results of the race. One of these scenarios uses the wind measured during the race. The reference race is R1 from the World Cup Series 2018 at Hyères, France.

To predict the minimal time path, this research reviewed the physical model for sailboats. Sailboats interact with water and air, while the seamanship is the one that controls it to reach any destination. Therefore, the sailboat is a rigid body that can move in a three-dimensional space. However, this research only considers the displacement in two dimensions.

Despite the similarities between Laser and yachts, one of the Laser adaptations is the addition of two coefficients to the sail's forces. These coefficients impact the Velocity Prediction Program/Polar (VPP) because it results from the interaction between the forces and moments generated by the wind and water when the sailboat is in motion. Moreover, the VPP shows the direction at which the Laser reaches its maximum velocity at alternative wind's velocities.

Wind is the main source of propulsion for sailboats and the seamanship uses the wind to maximize the velocity of the sailboat to adjust the direction and reach the target. Similarly, this algorithm uses the wind properties to find the path with the minimal time. The wind model is a forecast of four-dimensional and it describes the wind using a grid to locate the wind's speed over an area. Its resolution is the size of the grid, the distance between points and the time step.

Public information about wind forecast has a resolution distant from the characteristics of the race. The wind model used here is a Weather Research and Forecasting model (WRF) with a grid resolution of 1km, a time step of 10 minutes and an initial height of 7.5 meters over the area between 41.66°N, 4.75°E and 44.45°N, 7.25°E coordinates. While the measured data provided about the wind is in a tabular layout with a sample rate of 20 Hz. at 5 locations around the area of the event.

This algorithm discretizes the area of the race to break the problem into multiple stages all connected and to have grid points for the wind model. Because of this, the technique used is dynamic programming with a heading angle direction given by the VPP.

The scenarios used within this algorithm include step time variations of the WRF wind model and the wind measurements from the race besides the values of the wind properties without spatial variation along the race area. Thus, the scenarios to simulate the optimal time path are constant and uniform wind, wind field with a spatial resolution of 1km and a time step of 1 hour and 0.5 hours, the WRF and the wind measurements from the race.

In conclusion, the leg-times, start points over the race-lines and directions of the paths compose the minimal time path. Two of these elements were predicted similarly as the winners using the WRF wind model with an error in the race-time for less than 5%. However, the direction of the paths was not predicted accurately for the upwind-mode legs. Even though, this was equivalent to the winner using the constant and uniform wind scenario where the percentage error of the race-time respect to the winner is about 7%. Although here, the direction of leg 2 is not similar to the winners.

The grid resolution of the wind model is important for the minimal time path estimation. Moreover, the racecourse must be within the area of the wind model to avoid the extrapolation of the wind properties and percentage errors (%) larger than 25%. respect to the winner for leg-times sailed under upwind-mode. When the race area is not within the wind model area, the average values of the wind properties produce trajectories with smaller errors in time and shape.

This research proposes to review the location of the CE because at shorter distances from the sea level, the current and height of the waves could generate frictional forces insignificant at higher distances. Finally, to review the influence of the position of the sail-man using a three-dimensional model with the presence of waves and current during downwind and direct wind modes.

PREFACE

This dissertation is the minimal time path for Lasers. The Laser is one of the classes from Olympic Sailing and it uses one of the smallest boats known as a dinghy. Sailboats cannot displace against the wind following a straight line, because of this it is the time related to the optimization of the path and not the distance.

This thesis has been written to fulfill the graduation requirements of the Master's degree in Bio-Mechanical Design at TU Delft. This project was undertaken at the request of *InnoSportLab*®, The Hague where I undertook an internship. My research question was formulated together with professor Sukanta Basu and the *InnoSportLab*®. At the beginning, the research was difficult since the application of wind models on sports and the research about the Laser's dynamics is limited. The investigation I conducted was extensive and took me to unfamiliar topics, fortunately, all these have allowed me to answer the question of this thesis. I would like to thank my supervisor for his guidance and support during this process. I also wish to thank professor Sukanta Basu for his cooperation and my academic counselor Evert Vixseboxse for his support.

When I pursued a Master in Science at TU Delft I remember what my parents used to tell when I was a kid *"Learning is a never-ending process"*. Learning is more than books and science. Learning is also a process about ourselves. I dedicate this thesis to my parents and siblings, my best friends. Thanks for your support and sincerity. You help me learn how to feel comfortable out of my comfort zone, how to feel close to you despite the distance and even in moments where I doubt about myself. Thank you, Dad, because I still have not heard from you *No* as an answer. During this time at the Netherlands I have learned and growth as person and as a student in such ways I have not expected. This process sometimes was challenging, hard to endure and despite these I always got your love and support. You always cheer me and question me when I need it the most.

Finally, I would like to thank all the friends I made during this time you make my days, without you, this process would not be as fun as it was. To my friends Blanca Mayo and Kika, thank you for all your time, conversations and comprehension, I know I always can count on you.

Gabriela Pardo Arteaga
Delft, March 2019

List of Symbols

The next list describes several symbols that will be later used within the body of the document

Abbreviations

$\overline{GM}$	Metacentric Height
$DSYHS$	Delft Systematic Yacht Hull Series
B	Center of buoyancy
CE	Center of Effort
CG	Center of Gravity
CLR	Center of Lateral Resistance
g	Gravity
VMG	Velocity Made Good
VPP	Velocity Prediction Program/Polar
SNAME	Society of Naval Architects and Marine Engineers
DOF	Degrees of freedom
NWP	Numerical Weather Prediction
WRF	Weather Research and Forecasting Model

Symbols

α_a	Angle of attack of the sail
β_{aw}	Apparent Wind Angle
β_{i_a}	Angle of Attack when *i=k, for Keel*, and when *i=r, for Rudder*
β_{ti}	True Angle when i=c for Current, and when i=w for Wind
$\ddot{\phi}$	Angular Acceleration in the roll axis (*X-axis*)
$\ddot{\psi}$	Angular Acceleration in the yaw axis (*Z-axis*)
Δ	Vertical Displacement of the boat, related with the buoyancy when the boat is static

δ_s	Sail Trim Angle
$\dot{\phi}$	Angular Velocity in the roll axis (*X-axis*)
$\dot{\psi}$	Angular Velocity in the yaw axis (*Z-axis*)
γ	Course Angle
κ	Exponent for wind velocity at different height [1/7,1/4]
λ	Leeway Angle
$\mathscr{R}$	Rotational Matrix
ϕ	Roll Angle
Ψ	Heading Direction of the sailboat over the horizontal plane
ψ	Yaw Angle
ρ_i	Density, when i=a Air, i=w Water
Θ	Heel Angle
θ	Pitch Angle
A_{keel}	Keel Surface Area
A_{rudder}	Rudder Surface Area
A_s	Sail Area
C_d	Drag Coefficient
C_f	Hydrodynamic Coefficient for the waterline length
C_t	Lift Coefficient
C_w	Hydrodynamic Coefficient for the hull shape
I_{ii}	Total mass moment of inertia in the *i-axis*
J_{ii}	Total add mass moment of inertia in the *i* direction
K_i	Rolling moment (X-axis) of the *i* element
M_i	Moment, when i=R Righting, i=c Crew, i=Y Yaw, i=P Trimming
M_{k-r}	Moment produced by the keel and rudder
$N_{\dot{\phi}}$, $K_{\dot{\phi}}$	Hydrodynamic derivatives due to rolling
N_i	Yawing moment (Y-axis) of the *i* element
$V_{a_{w,c}}$	Apparent Velocity due to wind and current

V_{aw}	Apparent Wind Velocity
V_{ti}	True Velocity when i=c Current, i=w Wind
$V_{tw,c}$	True Wind Velocity affected by true current velocity(V_{tc})
V_{tw}^{b}	True Wind Velocity respect to the sailboat
W_c	Weight of the crew
$X_{V\dot{\psi}}, Y_{\dot{\psi}}, N_{\dot{\psi}}$	Hydrodynamic derivatives of the hull due to yawing motion

CONTENTS

1

Introduction

In recent years, it has increased the integration of new technologies on training programs of professional athletes. Not only to enhance their motor skills, and to prevent injuries but also to achieve better results with less time and in a more efficient process. For example, the use of information that specifies the conditions or parameters that deliver the best results to win, have gained more popularity within the sports community.

The involvement of science like kinesiology, movement science, and engineering should not focus only on the athletes but also in the elements and factors, they cannot control and impact their results, [1]. One of these sports is sailing where athletes have to guide the sailboat under different weathers and where time is crucial to winning the race.

During sailing races such as coastal, inland waterways and lakes, regulations do not allow that athletes get any information from coaches, especially during the race. Because of this, their preparation and training are not only physical. During training, they gain experience to prepare for weather changes and improve their maneuvers to perform better at races and therefore competitions.

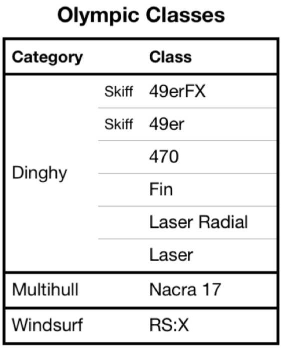
This section presents the research problem, its relevance, and objectives. For this, the first part describes briefly the Olympic sailing classes and courses followed by an explanation of the information considered by coaches and athletes. The information defines the plan and the strategy before the competition takes place which most of the time refers to the weather forecast. These data provide a framework to set the assumptions and limitations considered in this research.

1.1. Olympic Sailing

The rule of one-design boat characterizes the Olympic Sailing Classes. The intention of this is to leave out the influence of the technology on the results and to focus on the abilities of the athlete to perform effectively. In fact, the scope for many types of research is on the athlete only. The sailboat to use, size of the hull, the number of sails, its area, and the size and gender of the crew determine the sailing class. Figure 1.1 shows the names and categories of the current classes that the International Federation of sailing defines as Olympic Classes. The list is subject to changes, and the Sailing Organization review it every 4 years after the Olympic Games.

The dinghy is not only the smallest and simplest sailboat, one sail and one hull but the most used on different classes. The differences between classes using dinghies rely on the length of the boat and sail's area, these variations impact the speed of the boat and its maneuverability. Furthermore, the number of crew members,1 or 2, and gender determine the dimensions on the hull and sails.

This research uses the dinghy as a reference boat because is the simple boat, one hull one sail, used extensively on the Olympic Classes. Another reason is because of its similarity physically and mathematically with yachts, and scientists have studied them since 1979 [2]. This allows to use approaches and information from them either to compare or validate the results estimated by this research.



Olympic Classes

Category		Class
Dinghy	Skiff	49erFX
	Skiff	49er
		470
		Fin
		Laser Radial
		Laser
Multihull		Nacra 17
Windsurf		RS:X

Figure 1.1: Olympic classes [3].

The competition in each class stands for many races and days where the configurations of the racecourse vary according to the environmental conditions. In each race, according to its arrival position sailors win points, the faster, the lower the score. The winner is the one with the lowest score. Under this competing format, athletes and coaches confront distinct scenarios for which they take different decisions. One of these decisions refers to its location at the starting line and another to the initial sailing direction. The importance of them depends on the configuration of the course respect to the wind's direction.

1.2. COMPETITION COURSE

The most common form of a sailing competition is the fleet racing where all participants race around a course and start simultaneously along a line. Here, not only is the reaction time important but also the direction that allows the maximum speed. Usually, two distinct courses take place in the Olympic classes. Figure 1.3a shows the singular trapezoid course defined by a separate start and finish line and 4 points around (buoys). These 6 elements define the legs of the competition, the first leg is the longest, and it is running against the wind.

The trapezoid course is the most completed format, the other formats are a partial representation, considering only some legs oriented respect to the wind. The other course is the windward/leeward, figure 1.3b this is simply a two-leg race orientated in such way that the first leg is sail against the wind and the second leg is sail with the wind.

The courses characteristics and conditions of the race are also subject to change every 4 years. For example, the current course is regulated as follows, the length and angles of each leg are in such way that the race has a duration for a maximum of one hour and the trapezoidal course is inside a grid size of 2km by 2km maximum [4]. The diameter of a circle expressed in nautical miles (nm) determines the course area. Figure 1.2 is an example of how participants received information about the sailing areas where the trapezoid racecourse will take place.

Furthermore, the regulations include the wind conditions refer to the average wind speed and its shift direction. The current regulation establishes that the race will not start if the average wind speed is less than 4 knots(kn)[2 m/s] or over 25 knots(12.86 m/s) along the entire course, with maximum wind shift of 10deg. When the wind conditions are not meet, the judges delay the race and if the race has already started, it is possible that changes on the course happens or the race could be abandoned [4]. This is an example of how important is the wind for sailing. However, to understand how the wind interacts with the boat and the athlete, this research review first the physics of sailing boats during motion.

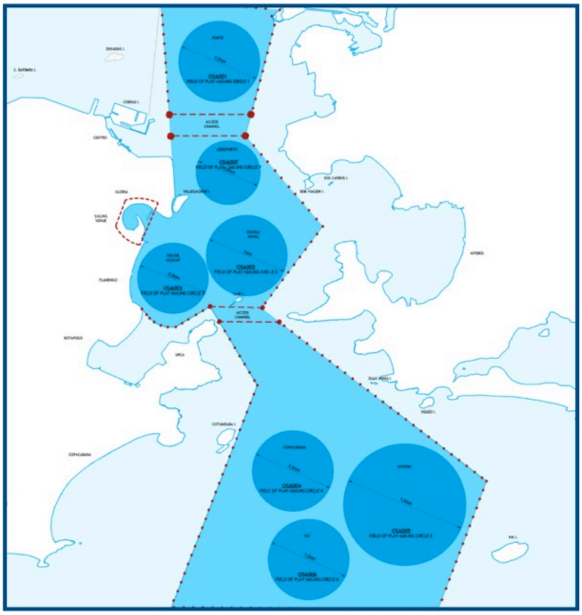
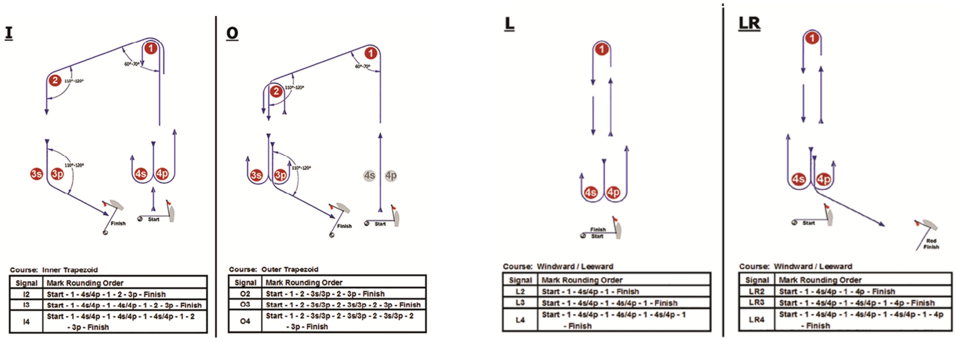

OSA001
FIELD OF PLAY SAILING CIRCLE 1
ACCESS CHANNEL
OSA007
OSA002
OSA003
FIELD OF PLAY SAILING CIRCLE 3
SAILING VENUE
GLORIA
CENTRO
FLAMENGO
BOTAFOGO
URCA
ACCESS CHANNEL
COPACABANA
OSA004
FIELD OF PLAY SAILING CIRCLE 4
OSA005
FIELD OF PLAY SAILING CIRCLE 5
OSA006
FIELD OF PLAY SAILING CIRCLE 6

Figure 1.2: Sailing Races Areas. Olympic Games Rio 2016 [5].

The direction of the boat regarding the wind determines the maneuvers and the trajectory that the athlete must follow. For example, the first leg of the trapezoid course runs against the wind and boats cannot moves towards it. The maneuvers used in this



Course: Inner Trapezoid

Signal	Mark Rounding Order
I2	Start - 1 - 4s/4p - 1 - 2 - 3p - Finish
I3	Start - 1 - 4s/4p - 1 - 4s/4p - 1 - 2 - 3p - Finish
I4	Start - 1 - 4s/4p - 1 - 4s/4p - 1 - 4s/4p - 1 - 2 - 3p - Finish

Course: Outer Trapezoid

Signal	Mark Rounding Order
O2	Start - 1 - 2 - 3s/3p - 2 - 3p - Finish
O3	Start - 1 - 2 - 3s/3p - 2 - 3s/3p - 2 - 3p - Finish
O4	Start - 1 - 2 - 3s/3p - 2 - 3s/3p - 2 - 3s/3p - 2 - 3p - Finish

Course: Windward / Leeward

Signal	Mark Rounding Order
L2	Start - 1 - 4s/4p - 1 - Finish
L3	Start - 1 - 4s/4p - 1 - 4s/4p - 1 - Finish
L4	Start - 1 - 4s/4p - 1 - 4s/4p - 1 - 4s/4p - 1 - Finish

Course: Windward / Leeward

Signal	Mark Rounding Order
LR2	Start - 1 - 4s/4p - 1 - 4p - Finish
LR3	Start - 1 - 4s/4p - 1 - 4s/4p - 1 - 4p - Finish
LR4	Start - 1 - 4s/4p - 1 - 4s/4p - 1 - 4s/4p - 1 - 4p - Finish

(a) Trapezoid Course

(b) Windward / leeward course

Figure 1.3: Types of courses [5].

condition are in figure 1.4a, this trajectory forms a zig-zag pattern that can start on the left- or on the right-hand side, each change on the heading direction is known as a tack.

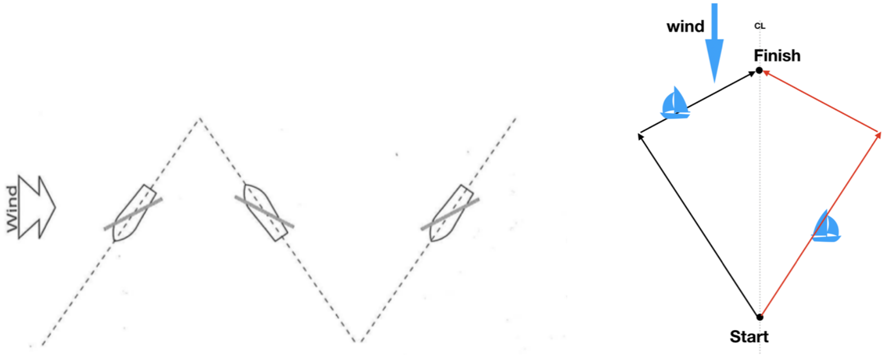
Even when the trajectory looks symmetrical, wind is not constant all the time and once the race starts, the wind should not shift more than 10°[4]. Because of this, athletes and coaches need to understand in advance how the wind shifts and its speed changes affect the path trajectory on each leg of the racecourse and its duration. Therefore the questions to answers are: *what direction athletes should take when the boat has to sail against the wind? Where is the path with the minimal time to follow?*.

To answer this question athletes and coaches rely on their prior knowledge, experience and available information to decide the starting direction. They base this knowledge on how familiar is the site to them. For example, top athletes for the Olympic Games train in the site at least one year or several months before the event. Moreover, the classification competitions take place at a same location thus coaches and athletes got an overview about the weather conditions. The geographical characteristics around the area of the course create patterns that athletes try identify to define a sailing strategy.

1.3. WEATHER FORECAST

The weather forecast is calculated by supercomputers and updated frequently, usually every one to two hours. Because of its complexity, different agencies and governments have developed models to predict the weather in global terms, thus local predictions can be made. These local predictions considers the global model and adjust it using the local measurements therefore the forecast model can be updating every hour with a grid resolution of 3km by 3km, at least, [7].

Uncertain weather is a typical condition that yacht competitions and maritime transportation have to manage. For yacht competitions, courses could take hours, days, weeks or even months, like the *Volvo Ocean Race* around the world. In the other hand, the path planning for vessels has been studied especially by maritime and logistics sciences. However, a smaller number of publications refers to yacht races and autonomous vehicles, and just a few of those publications refer to Olympic Classes only. This means that



(a) Tacking Maneuver [6].

(b) Tacking maneuver

Figure 1.4: Tacking maneuver against the wind.

the studies focus on the boat's motion but not on how to sail during a race.

Olympic races have a duration of maximum one hour inside a grid of size 2km by 2km while the local forecast is updated every hour within a grid of 3km by 3km. Because of this, the information that coaches and athletes can access previous to the race does not match their needs at first instance. Short and long racecourses are different. Short courses, for example, are more sensitive to random fluctuations [8]. Furthermore, the initial direction to take could be the optimal just for a couple of minutes but not for the whole leg.

Time on Olympic sailing classes is critical to winning, but at the same time, a higher resolution to represent the changes over time and space for the weather requires more computational effort for weather models and for the estimation of the minimal time trajectory. The *impact of the size for the resolution on the race time is unknown*. In the other hand, it is also unknown *how small this resolution must be*, to be significant for the minimal time path.

1.4. The Aim of the Research

This research proposes an algorithm to model sailing trajectories shaped by wind for Olympic Classes and compare wind models that have different time step sizes and spatial resolutions. Thus this research can answer the next questions, *Where is the optimal sailing route shaped by wind for the Laser (Olympic class) when the time and spatial resolution is constant for every hour or when it changes every 10 minutes. Which resolution is the most representative and provides accurate results?*. This study suggests that the sensibility of the optimal route due to changes in the wind field and starting direction can define zones and can shape the trajectory for minimal time paths.

Experiments will demonstrate the effects of four different time steps and spatial resolutions. The experimental result, provided by this optimization algorithm, will show the trajectories and its duration, by leg and in total. Besides these results can be compared with the measurements from a selected race. This recorded data serves as a reference for the experimental to quantify the error, due to the uncertainty from the wind model and other unknown factors, not only in the shape of the path-trajectory but also on the duration of each leg.

1.5. TERMS, CONDITIONS, AND LIMITATIONS

The algorithm developed is limited to dinghies, the smallest and the most used boat on the Olympic classes. More specific, the Laser Class with motion over the XY plane. The validation of the algorithm for the 2D model proposed results from the comparison between the results of the simulations (experimental results) against the results from the Laser race at Hyères, France during 2018.

The experiments to see the effects of the time step on the time and trajectory are four wind fields, first constant wind intensity and direction along the area of the route. The second, third and fourth scenarios, consider a forecast wind field as time-space-dependent variable changing every 60, 30 and 10 minutes, over 1 km by 1km grid size without current and neglecting the wave disturbances and twisting effects on the sail. The last test uses the wind measurements, taken during the race, to develop a wind field. The sample rate of the measurements is 20 Hz from five locations around the area for the whole event. The algorithm developed is implemented on MATLAB® and the optimization tool used is [9].

Olympic Classes like the dinghy is not a widespread research topic. Besides most of the findings related to path optimization for boats are related to cargo ships and vessels, where the aim, most of the time is to minimize fuel costs because these ships can navigate at a constant speed. A smaller amount of studies related to sports are focus to yachts and the influence of the wind is not the same as for the Olympic Classes.

This research does not consider the movement of the sailor as a variable since it assumes that the athlete is skilled enough to keep the equilibrium of the boat and maximize speed without further complications, this assumption is to other studies. For the crew's weight, the simulations use the weight proposed by [10], this weight is between 55 to 70 kg.

1.6. REPORT STRUCTURE

The report is set as follows: chapter 2 describes to the physics and dynamics models for the sailboat, basically, the forces and equations that govern its motion. The Wind model characteristics, its requirements for sailing and how it is integrated into the algorithm is reviewed in chapter 3. In chapter 4, the time optimization algorithm is explained as well as how the model is implemented with the required modifications for the purposes of this study. The results are analyzed in chapter 5 and finally, the conclusions and recommendations are in chapter 6.

2

THE PHYSICS BEHIND SAILBOATS

Sailing boats are propelled mainly by wind. However, they move through the water. This means that sailboats move through two different fluids: water and wind. The mechanics of sailing have been known since the 1950s. Marchaj in 1979 did a review about them and add information which is still being used for yacht design [2].

This chapter will focus on the motion of the sailboats and the physics behind it. The forces that interact in the equilibrium and during motion. How the athlete takes part in this model and what other considerations are required to set-up the equations of motion. Since sailboats are governed by similar equations some adjustments are required to differentiate between yachts and lasers, and these adjustments are explained on detail on later chapters.

The equations of motion facilitate the identification of variables that can be used as a parameter to model the trajectory as well as to identify its limitations. These considerations are the requirements to optimize the trajectory and obtain a minimal time path.

2.1. THE INTERACTION BETWEEN THE SAILBOAT, WATER, AND AIR

The origin of the forces and momentum depend on the interaction of the elements from the sailboat with the two mediums, water and air. Some of these forces are easier to identify, like the forces acting above the water surface which are produced by the wind and its interaction with the sails. These forces have to be balanced by the forces beneath the same surface. In this case, the water interacts with the hull, rudder, and keel. Therefore by adjusting the sail and the rudder, the only elements controlled by the athlete, the sailboat can hold a steady course.

Philpott explains how the different components and parts of the sailboat, interact with the surroundings and how they are used to control and attain the equilibrium during motion, [11]. Figure 2.1 shows the most common elements and where are they located. Some of those elements can be manipulated by the seamanship, this means that the variables concerned can be controlled and therefore they are known as control variables [11].

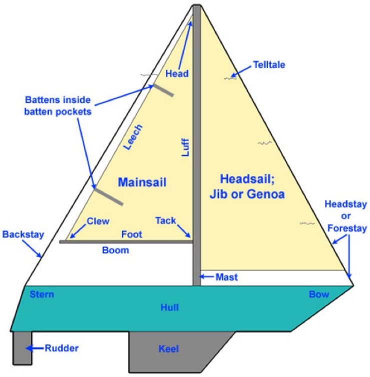


Figure 2.1: Common sailboat terms [12].

In order to steer a boat, the seamanship has to control the angle of the rudder, this interacts directly with the current, the direction obtained is called *heading* and these two, rudder and current, generate forces that influence the boat to *yaw*.

Due to the wind direction, mainly, the boat slips sideways and this effect is known as *leeway*. The difference in course comparing with the heading is expressed as *leeway angle* (λ). The sails adjustment is known as trim. When the trim reduces the area of the sail then the seaman is *reefing*, most of the time this term refers to the sails when its size is changing. Reefing under sail allows the seamanship to control the wind intensity.

The wind over the sails generates a force and an angle called *heel angle*, indicated in the figure 2.4B, this decreases the driving force. Under those circumstances, a moment is generated and to neutralize it, the seamanship generates a *righting moment* (M_R) by standing on the windward side of the boat to produce it[11]. As a result of these forces, the velocity could be optimal or not. [13] relates the factors and forces proposed in[11] in terms of forces and resistances, indicating how the dynamics of each of the mediums, water, and air, interact to keep the balance (and be capable to maximize the boat's speed). Therefore, the forces and resistances are related as showed below:

- Aerodynamic driving forces = Hydrodynamic resistance;
- Aerodynamic side force = Hydrodynamic side force;
- Aerodynamic heeling moment=Hydrodynamic (static) righting moment.

An important assumption made by [11] and [13] to keep the analysis of the boat in 2 dimensions is that vertical forces are always in balance. The same also hold for the pitching moment. Figure 2.4 **c** shows the only forces that act when this assumption is made, additionally two angles are shown, one of then refers to the wind.

Translation	Axis/Position
- Surge	x
- Sway	y
- Heave	z
Rotation	
- Roll	x
- Pitch	y
- Yaw	z

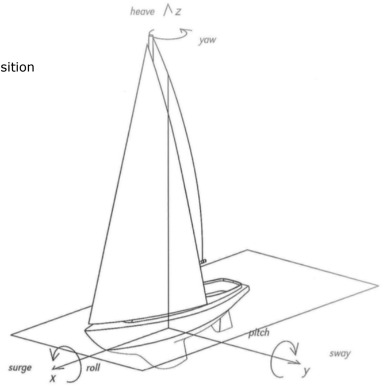


Figure 2.2: Degrees of freedom of a boat, clockwise reference system *XYZ* [14].

2.2. PLANES OF MOTION

Sailing boats are considered rigid bodies that can move in a three-dimensional space, figure 2.2 shows the six fundamental types of motion or the degrees of freedom (DOF) with the names and axis where they are referred. These motions are three translations and three rotations.

This figure also shows the water surface which is represented by the plane *XY* and the orientation of the sailboat with respect to the positive direction of the three axes which follow the right-hand orthogonal system. Thus, *X-axis* is positive in the direction of the motion, *Y* is positive to left side of the sailboat and *Z* is positive upwards. In the case of the *Y-axis*, the negative direction or right side of the sailboat is known as starboard. Besides, for future references, the speed of the boat is along the *X-axis*. These six DOF correspond to three forces and three moments which are going to be explained in the next section.

The sailboat interacts between two fluids and they generate forces and resistances in both mediums. It is important to know how the elements are named and which words are related with a specific axis and which elements interact on it and how it is conserved when the sailboat moves from one point to another.

2.3. HYDRODYNAMIC AND AERODYNAMIC FORCES AND MOMENTUM

The static and dynamic balance of any type of boat is based on Newton's second law. When the boat moves there is a dynamic equilibrium generated mainly by the winds' velocity. This force is counterbalance by a resistance generated by the sailboat through the water. The next equations will show that the total aerodynamic force (F_A) is equal and opposite to the total hydrodynamic force ($F_{H_{TOT}}$).

Motion/Rotation	Force/Moment	Linear/Angular vel	Position/Angles
Surge(X-axis)	X	u	x
Sway (Y-axis)	Y	v	y
Heave(Z-axis)	Z	w	z
Roll (X-axis)	K	p	ϕ
Pitch (Y-axis)	M	q	θ
Yaw (Z-axis)	N	r	ψ

Table 2.1: SNAME's notation for motion components [15]

From section 2.2, a set of vectors can be defined to represent the position, velocities, and forces. The notation used here is similar to the one used by the Society of Naval Architects and Marine Engineers (SNAME) because different authors use different notation, this work is intended to stay close to the norm shown in the table 2.1, this notation refers to the body-fixed reference frame.

2.3.1. WIND AND THE VELOCITY TRIANGLE

In sailing, the wind is characterized by its speed and direction and it is defined as *true wind velocity (V_{tw})* and *true wind angle (β_{tw})*. Because it also interacts with the water surface, in some cases its intensity depends on the height where it was measured, the equation to estimate its value at a different height is equation 2.1. According to [16] the exponent κ has a value between 1/7 and 1/14, and the reference height for measurements is 10 meters above the sea level.

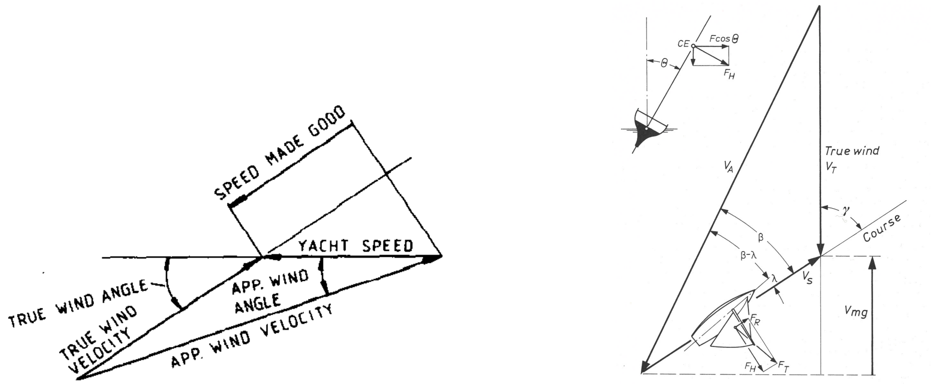
$$V_{tw}(Z) = V_{tw}(Z_{ref}) \cdot \left(\frac{Z}{Z_{ref}}\right)^{\kappa} \tag{2.1}$$

The steady motion of the sailboat not only depends on the balance of forces but also on the relation between velocities from the boat, and wind, mainly. This interaction is represented by the velocity triangle shown in figure 2.3a. The triangle introduces the apparent wind velocity (V_{aw}) and angle (β_{aw}).

The V_{aw} and β_{aw} result from the vector summation of the true wind (V_{tw}) and sailboat's (V_{boat}) velocity, 2.2. A more complete estimation of them uses the heel angle (Θ), equations 2.3 and 2.3 show the relation to calculate them. These equations include the heel angle because V_{aw} is used to calculate sail coefficients related to forces and they are specified at the center of effort (CE), its location is about 40% of the mast height. The β_{aw} incorporated the leeway angle (λ), and this value is usually less than 6°[11],[16]. Subsequently, the course angle γ is complementary to the β_{aw} when λ is not bigger than 6°2.3b.

$$\overrightarrow{V_{aw}} = \overrightarrow{V_{tw}} - \overrightarrow{V_{boat}} \tag{2.2}$$

$$\beta_{aw} = tan^{-1}\left(\frac{V_{tw} sin\beta_{tw} cos\Theta}{V_{tw} cos\beta_{tw} + V_{boat}}\right) \tag{2.3}$$



(a) Velocity triangle [13].

(b) Angles and course direction [2].

Figure 2.3: Velocity Triangle and angle directions of wind and λ [2], [13].

$$V_{aw} = \sqrt{(V_{tw}sin\beta_{tw}cos\Theta)^2 + (V_{tw}cos\beta_{tw} + V_{boat})^2} \tag{2.4}$$

2.3.2. EQUILIBRIUM EQUATIONS

In the static condition, equilibrium is reached when the summation of all the forces and momentum equals zero. In figure 2.4 these forces are located according to the planes where they act, the names of them and the fluid that drives them. In addition to the forces and momentum, there are 4 angles to consider. Another observation is how the position and weight of the athlete are incorporated in the equilibrium equations.

The equations of forces(F) by plane in equilibrium are:

$$\text{(Surge, x axis)}\ \ F_R = R \tag{2.5}$$

$$\text{(Sway, y axis)}\ \ F_{H,lat} = F_{S,lat} \tag{2.6}$$

$$\text{(Heave, z axis)}\ F_V = F_{VW} \tag{2.7}$$

And those for the momentum (M) in equilibrium are:

$$\text{(Roll, x axis)}\ \ M_R = M_H \tag{2.8}$$

$$\text{(Pitch, y axis)}\ M_{PA} = M_{PW} \tag{2.9}$$

$$\text{(Yaw, z axis)}\ M_{YW} = M_{YL} \tag{2.10}$$

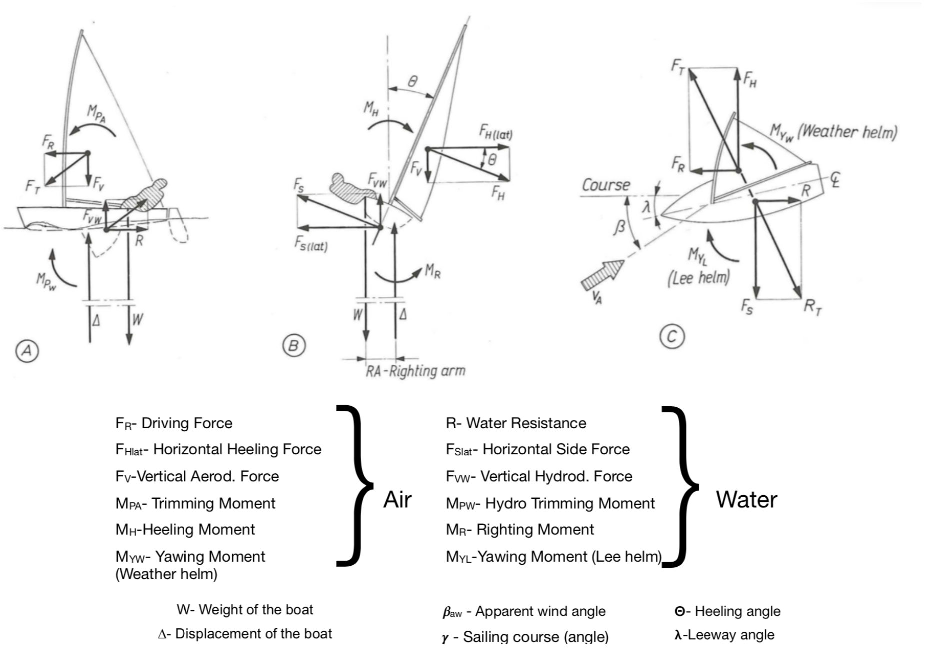


Figure 2.4: Equilibrium of forces and moments in steady-state sailing condition [2]

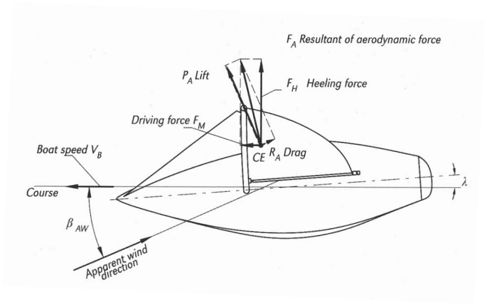


Figure 2.5: Total Aerodynamic Forces [14]

2.3.3. Aerodynamic Forces and Resistances

The force that drives motion of the sailboat is the driving force(F_R), figure 2.4 *B* shows the force that causes the drift (heel) of the sailboat called the heeling force (F_H). The motion of the sailboat happens when F_R beat the hull resistance (R), while in the ZY plane the balance of the forces is realized when F_H equals the hydrodynamic side force (F_S), which is produced by the effect of the water over the hull. Then:

$$\begin{gathered}F_A = F_R(\parallel V_{aw}) + F_H(\perp F_R)\\ F_R = F_R(V_{aw}, \beta_{aw}, \Theta)\\ F_H = F_R(V_{aw}, \beta_{aw}, \Theta)\\ F_{S_{lat}} = F_S(V_{boat}, \lambda, \Theta)\\ R = R(V_{boat}, \lambda, \Theta)\end{gathered} \tag{2.11}$$

Due to the dependency on velocities, F_R and F_H generate drag (D) and lift (L) forces acting normal to the center plane of the hull and mast and they are integrated into the forces, figure 2.5 shows the aerodynamic forces, according to [11] and [16] hence these forces are:

$$F_R = Lsin\beta_{aw} - Dcos\beta_{aw} \tag{2.12}$$

$$F_H = (Lcos\beta_{aw} + Dsin\beta_{aw})cos\Theta \tag{2.13}$$

D and L depends not only on the sail area (A_s), V_{aw}, and fluid density (ρ_a), in this case air, but also on coefficients which depends on the trim and flatness of the sail [11], [17], [18]. These variables are under the control of the seamanship. D and L are expressed in those terms as follow:

$$L = qA_sC_t \tag{2.14}$$

$$D = qA_sC_d \tag{2.15}$$

$$q = \frac{1}{2}\rho_a V_{aw}^2 \tag{2.16}$$

$$C_d = C_d(\beta_a, trim, flatness) \tag{2.17}$$

$$C_t = C_t(\beta_a, trim, flatness) \tag{2.18}$$

where:

- ρ_a air density approx. 1.225 kg/m^3.
- A_s is the area of the sail.

C_d and C_t values are obtained from tables or graphics and their range is over (0,1), later on, this is going to be explained in more detail.

F_R is the total force applied on the sail which can be decomposed in 2 forces, a driven force F_{ms} and a side force F_{ss} expressed in the terms mentioned as before:

$$F_{ms} = (Lcos\beta_{aw} + Dsin\beta_{aw})cos\Theta sin\lambda + (Lsin\beta_{aw} - Dcos\beta_{aw})cos\lambda \tag{2.19}$$

$$F_{ss} = (Lcos\beta_{aw} + Dsin\beta_{aw})cos\Theta cos\lambda - (Lsin\beta_{aw} - Dcos\beta_{aw})sin\lambda \tag{2.20}$$

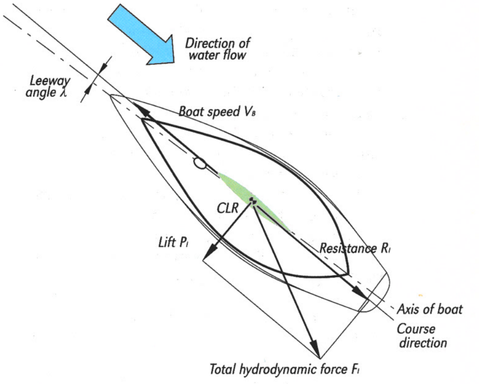


Figure 2.6: Total Hydrodynamic forces. [14]

2.3.4. Hydrodynamic Forces and Resistances

$F_{H_{TOT}}$ is equivalent to F_A with the difference that they result from the interaction with the water. In this case, R, a drag force (resistance) oppose the motion of the sailboat as shown in figure 2.6. The horizontal side force $F_{S_{lat}}$, is a lift force acting over the hull and keel. Most of the information related with the modeling of the hull and keel is based on experimental data. The drag generated by the hull is the result of the upright, heeled and induced resistances [11].

$$F_{H_{TOT}} = R(\parallel(-V_{aw})) + F_{S_{lat}}(\perp R) \tag{2.21}$$

The upright resistance is produced by the hull drag when λ is zero. It is constituted by the friction generated by the water viscosity and wave drag, dissipation of energy in form of waves due to the shape of the hull. The formula to calculate these two frictional forces, equation 2.22 and 2.23 is similar to the one from classical hydrodynamics, the difference is the coefficients. C_f depends on V_{boat} and its waterline length while C_w depends on the hull shape and Froude number (Fr), which considers the V_{boat} and length of the sailboat.

$$F_f = \frac{1}{2}\rho_w C_f A_w V_{boat}^2 \tag{2.22}$$

$$F_w = \frac{1}{2}\rho_w C_w A_w V_{boat}^2 \tag{2.23}$$

where:

- ρ_w water density approx. 1000 kg/m^3.
- A_w is the wetted surface area of the sailboat

The induced and heeled resistances are combined and related with the heel and λ, and its formula also depends on V_{boat}. Due to its complex derivation and the existence of many different versions the expression is going to be set as: $F_i(\lambda, \Theta, V_{boat})$, for the purpose of this work. Then R due to hydraulic resistances is calculated as:

$$R = F_f + F_w + F_i \tag{2.24}$$

In the case of $F_{S_{lat}}$, it depends also on three resistances generated by the hull, the lifting surface of the rudder and keel. These two last are the more significant and they are also perpendicular to the velocity. This total side force (*S*), on the water plane, depends on the plan area of the keel and rudder and in two coefficients, respectively. Thus, it is estimated as:

$$S = \frac{1}{2}\rho_w V_{boat}^2 (C_{rudder}A_{rudder} + C_{keel}A_{keel}) cos\Theta \tag{2.25}$$

.

The shape, of the rudder and keel, in conjunction with the angle of attack, has implication on each coefficients. The angle of the rudder (β_r) is the angle it makes with the center line of the sailboat. Each angle of attack (β_{i_a})is determined as:

$$\beta_{r_a} = tan^{-1}(cos\Theta cos\lambda + \beta_r) \tag{2.26}$$

$$\beta_{k_a} = tan^{-1}(cos\Theta cos\lambda) \tag{2.27}$$

$F_{S_{lat}}$ in equilibrium can be estimated by using F_H, this generates a reaction force below the water surface, and this is know as horizontal side force $F_{S_{lat}}$ and it is determined as follow:

$$F_{Slat} = F_H cos\Theta \tag{2.28}$$

2.3.5. MOMENTUM AT THE SAILBOAT MODEL

The total aerodynamic force F_A is applied at *CE*, above the waterline and located over the sails area. The total hydrodynamic force F_{H_Tot} is at the center of the lateral resistance (*CLR*), defined below the waterline. The moments generated over the sailboat depends on these four variables. For example, the *righting moment (*M_R*)*, figure 2.4 *B* is counterbalanced as next:

$$M_R = F_H(CE - CLR)_z = W \cdot RA \tag{2.29}$$

where:

- $(CE - CLR)_z$ is the heeling arm or the vertical distance between *CE* and *CLR* when F_H is perpendicular to it.
- W = weight of the boat.
- RA = Righting arm or the horizontal distance between the *W* and the *Z* axis.

This expression does not take into account the weight of the crew. The moment generated by the crew weight is given by the weight of the crew (W_c)) and its relative position to the centerline of the boat, this is expressed by a variable known as y_c. This variable has a range value of [-1,1], and if the crew is in the centerline then its value is zero. Another factor to considers is Θ. Thus, the moment generated by the crew according to Philpott, [11] is:

$$M_c = y_c W_c Y_{max} cos\Theta \tag{2.30}$$

$$Y_{max} = \frac{1}{2} beam(width)_{sailboat} \tag{2.31}$$

and the *total righting moment ($M_{R_{TOT}}$)* is:

$$M_{R_{TOT}} = M_R(V_{boat}, \Theta, \lambda) + M_c \tag{2.32}$$

The *yaw moment* (M_Y) causes the rotation on the *Z-axis* and depends on the rudder angle (β_r), most of the time, since it can shift the *CLR*, another way to change this distance is by trimming, which changes the area of the sail, therefore, the *CE* shifts its location. If the sailing motion is steady, then this moment is zero. Otherwise, it can be calculated by the hydrodynamic resistance generated by three the elements on the boat: from the hull, the keel and the rudder. Each element generates a side force and these forces are compensated by the sail force [11], [16]. The moment generated by these three forces (M_Y), and its equation is:

$$R_{hull} = R + F_H = F_f + F_w + F_H \tag{2.33}$$

$$M_{hull} = R_{hull}(CLR_y cos\lambda sin\Theta - x_y sin\lambda cos\Theta) \tag{2.34}$$

$$M_{k-r} = \frac{1}{2}\rho_w V_{boat}^2 (C_{rudder} A_{rudder} x_r + C_{keel} A_{keel} x_k) cos\Theta cos\lambda \tag{2.35}$$

$$M_{sail} = x_s F_{ss} + z_0 r sin\Theta F_{ms}) cos\lambda \tag{2.36}$$

The 2.34 uses x_y which refers to the distance of the *CLR* from the *Z-axis* (*yaw axis*) while in 2.35 x_r and x_k are the lever arms of the rudder and keel, respectively. Finally, in 2.36 x_s is the distance from the yaw axis to the *CE* and $z_0 r$ is the lever arm of it, where z_0 is the height of the *CE* and r is the reefed/trim proportion of the sail. As mentioned in section 2.3.1 λ is small which allows *cos* and *sin* functions to be simplified by linear approximations.

The *trimming moment* (M_P) arise from gravity (g) and buoyancy forces, the displacement of the boat (Δ) is related with V_{boat}. For example, at high velocities, the lift forces are added to the buoyancy forces at the front sections of the sailboat. In case of rough water, however, the pitching motion reduces the speed and it is compensated by the aerodynamic and hydrodynamic forces, [16].

The equilibrium of forces and moments of the sailboat results from the interaction between the forces generated by the air, or by water or by both into different parts of the sailboat. These interactions in some cases are sophisticated, in the sense that their values depend on coefficients related with the shape of the element where the force is generated, the complexity is because some of the elements are assumed to be rigid, however, and at some speed this assumption does not apply.

The steady motion is reached when all these forces are in equilibrium considering the wind and its direction. Due to the complexity of the relations mentioned before the equilibrium solution is not always unique. Because of this, it is possible to sail at different directions and to reach a place even when it is on the windward side of the sailboat.

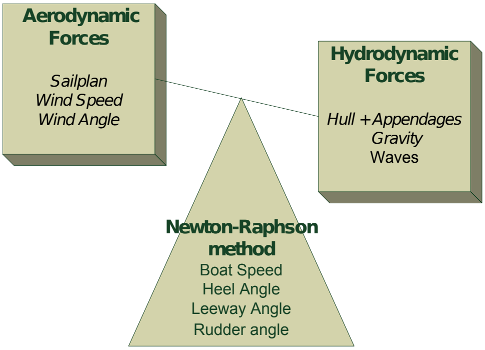


Figure 2.7: VPP Force Balance Representation [19]

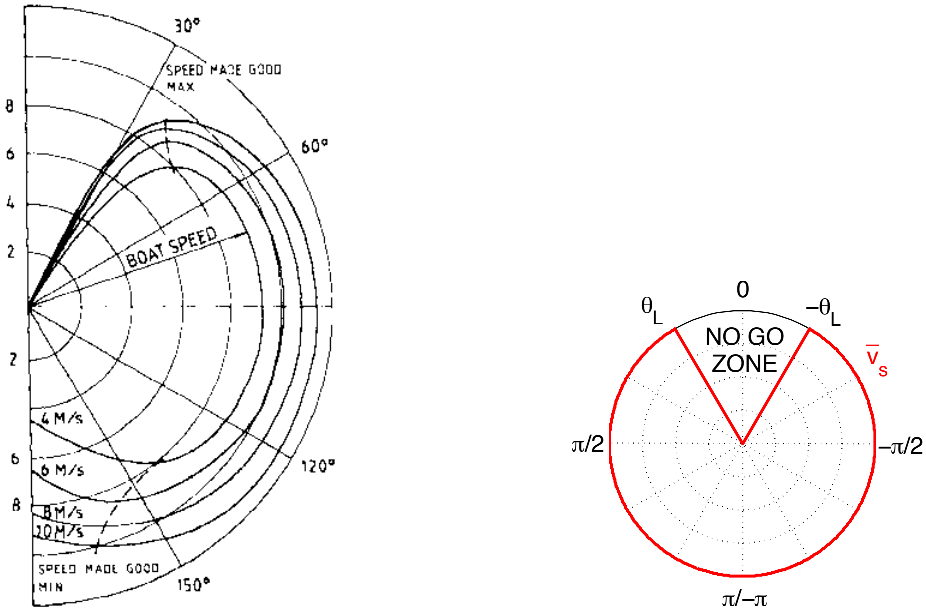
2.4. VELOCITY PREDICTION POLAR

In 1979 scientists developed the velocity prediction program (*VPP*) with the objective to predict the sailboat speed and direction for any wind condition, magnitude, and direction [13]. The results can be interpreted easily using a polar diagram or by reading its results in the form of tables. The kinetics of the sailboat explains how the motive forces from sails equal the hull resistance and how side forces from keel and rudder equal the sail side forces. In other words, how the aerodynamic forces and momentum are counterbalanced by the hydrodynamics of it. These relations and results are obtained from the static equilibrium equations mentioned in section 2.3.2.

The VPP contains the solutions of the equations to accomplish this balance and it requires not only to solve the hydrodynamic and aerodynamic equations but also to know the properties of the sailboat related with its stability, such as λ [13], [20]. The information required to calculate a VPP comes from different sources and all the variables involved can be classified in one of the four categories or groups. Following [11] these groups are:

- design ($_d$): describes the size and shape of the sailboat and its elements.
- environment ($_e$): describes the wind and current in which the sailboat will perform.
- control ($_{x_c}$): these are setting variables that can be adjusted within some limits or constraints by the seamanship, such as β_r, y_c, f, and r.
- behavior ($_{y_b}$): these variables describe the motion or condition of the motion at a given time or due to a given environmental condition. Examples of this type or variables are: λ, β_{tw}, Θ, and the V_{boat}

The remaining variables can be grouped as *auxiliary* variables ($_{aux}$). These variables describe transitional stages or intermediate calculations, such as V_{aw}, D, L, M_c, M_{sail}



(a) VPP plot for true wind angles from 0 °to 180°and true wind speeds from 4 to 10 m/s (7.77-19.44 kn) [13].

(b) Polar Curve of the Propulsion System [21](Full VPP representation).

Figure 2.8: VPP diagram

among others. The arrangement of these groups results in at least 22 simultaneous non-linear equations which have to be solved by specifying a performance criterion and optimizing the y_b variables.

Because VPP plots are symmetric only half of it is usually represented as shown in figure 2.8a. The polar diagram indicates the wind direction or *true wind angle* at 0°, V_{boat} is defined by the radius size of the concentric half circles, the straight lines indicate the direction of the boat from the wind direction. Last, the wind speed is the group of lines that form a half heart shape. The intersection of these lines with the direction lines indicate the maximum velocity that a sailboat can attain.

By using the VPP not only the direction of the maximum sailboat speed can be identified but also it shows the direction where the speed is the slowest. The angle range of this speed or speeds corresponds to the concave curve of the true wind, the area is defined as the *no-go-zone* and it represents the set of directions that should be avoided to stay away from the irons [21],[6]. This means that under these directions the sailboat motion is minimal, its propulsion is not enough or it is minimal to outweigh the resistance derived from the hull and sail.

The performance criterion most used is to maximize V_{boat} except when the sailing is towards the wind, upwind condition or sailing to windward. In this case, the performance criterion change from velocity to the distance that can be travel during a certain time. This criterion is known as the velocity made good *(VMG)* and it indicates where the sailboat is on the space from a reference and how is its motion [13], [2] [11].

VMG is determined by equation 2.37. This relation can also be found in the velocity triangle, figure 2.3a, a more detailed geometry about this is the figure 2.9. In the figure it is possible to identify how the β_{tw} and λ interact to get the β_{aw} therefore, how is the distance from the origin or reference point. The figure also indicates the optimal angle, this angle indicates the maximun velocity or the minimal time to reach a distance with the shortest possible time. The VMG also indicates how different angles perform under the same time interval.

$$VMG = \mid V_{boat} cos \beta_{tw} \mid \tag{2.37}$$

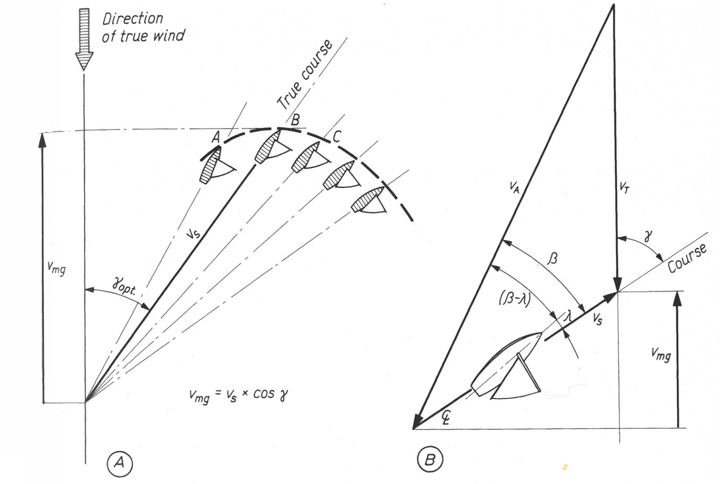


Figure 2.9: Definition of VMG. A. VMG at different angles. B. Velocity triangle including the leeway angle [2].

The VPP is obtained by balancing the equations from previous sections, from equation 2.5 to equation 2.36. It derives the V_{boat} at different wind conditions and at different directions respect to the wind.

These equations are usually not solved in its six DOF. Because it was assumed that the vertical forces and moments are always in equilibrium, as was mentioned by [13], [14] and section 2.1. The VPP partially solves the equations, this means that the result provided only applies on 2D, particularly in the plane *XY*. Thus it only considers the rolling moment and longitudinal forces, because changes on the *Z* plane are neglected. Some of the variables related with the *Z-axis* are still considered to solve the system of equation, one of these variables is Θ.

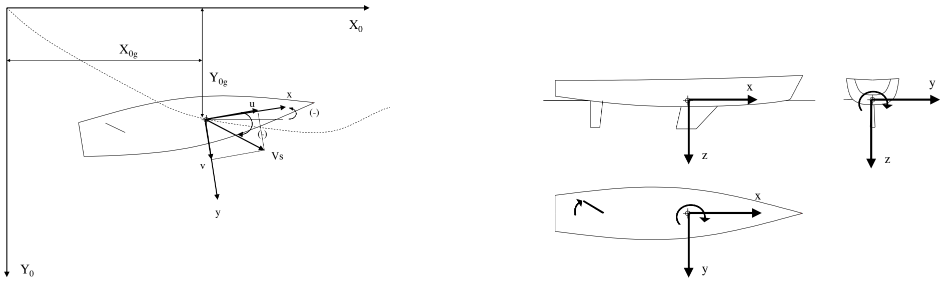
In this section, it has been identified how the variables are categorized, most importantly which variables can be adjusted by the seamanship and which are defined by the sailboat and weather.

2.5. EQUATIONS OF MOTION

The motion of the boat from one point to another occurs in the XY plane. In the previous sections, the different components like forces generated by the wind and water were explained and how the seamanship or athlete compensate these forces by interacting with sailboat via sails,β_{tw}, and rudder, λ, mainly. In section 2.3 it was mentioned that pitching moment and sway forces (vertical forces) are always in equilibrium, which means that heave and pitch motion can be omitted and the motion analysis only occurs on the XY plane within four DOF. These motions are the surge, sway, yaw, and roll.

In 2004, de Keuning. et. al [22] propose a mathematical model to describe the motion of sailboats or *tacking maneuvers* integrating the information provided by the Delft Systematic Yacht Hull Series (DSYHS), in this study the estimation of the coefficients can therefore be defined from this model without the need for experimental data for a specific sailboat.

In the model, the main change refers to its coordinate system, figure 2.10 shows that the coordinate system is at the center of the sailboat, the *Z-axis* is positive orientation when it points downwards and the *Y-axis* is pointing at starboard [22].



(a) Motion depict by the mathematical model.

(b) Angle's location and axis orientation according to the coordinate system.

Figure 2.10: Motion depiction of the mathematical model bestow by the coordinate system [22].

The next equations or Euler's equations for sailboats motion described by [22] define the total forces applied in the sailboat element corresponded to the main axis.

The forces on the X and Y axis are:

$$X_U + X_{hull} + X_{rudder} + X_{sail} = m_T(\dot{u} - v\dot{\psi}) \tag{2.38}$$

$$Y_{hull} + X_{rudder} + X_{sail} = m_T(\dot{v} - u\dot{\psi}) \tag{2.39}$$

And the momentum on the X and Y axis are:

$$K_{hull} + K_{rudder} + K_{sail} + K_{stability} = I_{xx}\ddot{\phi} \tag{2.40}$$

$$N_{hull} + N_{rudder} + N_{sail} = I_{zz}\ddot{\psi} \tag{2.41}$$

where:

- m_T = the total mass of the sailboat not including crew.

- u = velocity along the X-axis.
- v= velocity along the Y-axis.
- ϕ = roll angle, related to Θ.
- ψ = yaw angle.
- I_{xx} = total mass moment of inertia in the roll axis (X.axis).
- I_{zz} = total mass moment of inertia in yaw axis (Z-axis).
- K = Rolling moment (X-axis).
- N = Yawing moment (Z-axis).
- X_U = is the hull resistance in the upright position.

It is important to clarify that ψ and ϕ are variables to describe the location of the sailboat over an area. They are related to Θ and λ from the kinematics of the sailboat. Because the motion described by Keuning et. al [22] includes waves effect there is a difference between ϕ and Θ. The heel angle Θ is the rotation from the Z plane in *flat water*; while ϕ, the roll angle, is Θ plus the rotation induced by the fluctuations on the front wave [23], [6].

$K_{stability}$ is the moment generated by the weight when Θ is not zero; the center of buoyancy (B) of the sailboat moves laterally and when its vertical line crosses the mid-plane of the sailboat creates a metacenter (M). The distance between CG and M is known as the metacentric height, $\overline{GM}$) [24] this displacement generates an oscillation and a mass moment that is calculated as follow:

$$K_{stability} = -mg\overline{GM}sin\phi \tag{2.42}$$

The rudder and the sail are sailboat elements controlled by the seamanship. The rudder produces hydrodynamic forces and momentum, as explained in section 2.3.5. And the weight of the crew generates a momentum according to the equation 2.30. This momentum can be included in the previous equations by assuming that the centroid of W_c is located in the center of gravity (CG) and the equations are:
Surge (X-axis):

$$X_U + X_{hull} + X_{rudder} + X_{sail} + X_{V\dot{\psi}}V\dot{\psi} = (m+m_x)\dot{u} - (m + m_y cos^2\phi + m_z sin^2\phi)v\dot{\psi} \tag{2.43}$$

Sway (Y-axis):

$$Y_{hull} + Y_{rudder} + Y_{sail} + Y_{\dot{\phi}}\dot{\phi} + Y_{\dot{\psi}}\dot{\psi} = \{(m+m_y)cos^2\phi + m_z sin^2\phi\}\dot{v} + (m+m_x)u\dot{\psi} + 2(m_z - m_y)sin\phi cos\phi \cdot v\dot{\phi} \tag{2.44}$$

Roll (X-axis):

$$K_{hull} + K_{rudder} + K_{sail} + K_{stability} + K_{\dot{\phi}}\dot{\phi} = (I_{xx}+J_{xx})\ddot{\phi} - (I_{yy}+J_{yy}) - (I_{zz}+J_{zz})sin\phi cos\phi \cdot \dot{\psi}^2 \tag{2.45}$$

Yaw (Z-axis):

$$N_{hull} + N_{rudder} + N_{sail} + N_{\dot{\psi}}\dot{\psi}$$
$$= \{(I_{yy} + J_{yy}) sin^2\phi + (I_{zz} + J_{zz}) cos^2\phi\}\ddot{\psi} + 2\{(I_{yy} + J_{yy}) - (I_{zz} + J_{zz})\} sin\phi cos\phi \cdot \dot{\psi}\dot{\phi} \tag{2.46}$$

where:

- m = mass of the boat without crew
- m_i = the added masses in i direction.
- I_{ii} = total mass moment of inertia.
- J_{ii} = total added mass moments of inertia.
  - i= x,y and z.
- $X_{V\dot{\psi}}$, $Y_{\dot{\psi}}$, $N_{\dot{\psi}}$ = hydrodynamic derivatives of the hull due to yawing.
- $N_{\dot{\phi}}$, $K_{\dot{\phi}}$ = hydrodynamic derivatives of the hull due to rolling.

The last equations determine the position of the boat and the assumption that the heel angle is small close to 0 ($\Theta \approx$ **0**)means that: X_{hull}**,** Y_{hull}**, and some terms of** M_{hull} **and** M_{sail} **are canceled, and therefore *yaw equation* can be neglected**. The *yaw equation* describes the balance of the boat in the yaw angle here it is assumed that the seamanship always keeps this balance. Moreover, competitions are regulated in such way that the safety of the participants is not compromised, this last restrict the V_{boat} and prevent unbalance situation on this axis.

With these assumptions the equations of motion can be simplified as:
Surge (x axis):

$$X_U + X_{rudder} + X_{sail} + X_{V\dot{\psi}} V\dot{\psi} = (m + m_x)\dot{u} - (m + m_y)v\dot{\psi} \tag{2.47}$$

Sway (y axis):

$$Y_{rudder} + Y_{sail} + Y_{\dot{\phi}}\dot{\phi} + Y_{\dot{\psi}}\dot{\psi} = (m + m_y)\dot{v} + (m + m_x)u\dot{\psi} \tag{2.48}$$

These two equations can be reorganized so to get the accelerations vector of the sailboat. The only decision variable identifies until now refers to $\dot{\psi}$, which is related to the β_{aw}, therefore, to λ. The equations 2.12,2.24 and 2.25 from section 2.3 are substituting these in equations 2.5 and in 2.48, then the previous equations becomes:

$$X_{TOT} = X_U + X_{rudder} + X_{sail} \tag{2.49}$$

$$X_{TOT} = Rcos\psi + F_R cos\psi + Scos\psi + \tag{2.50}$$

$$Y_{TOT} = Y_{rudder} + Y_{sail} \tag{2.51}$$

$$Y_{TOT} = F_R sin\psi + S sin\psi \tag{2.52}$$

$$\dot{u} = \frac{X_{TOT}}{m+m_x} + v\frac{m+m_y}{m+m_x}\dot{\psi} + \frac{X_{V\dot{\psi}}}{m+m_x}V\dot{\psi} \tag{2.53}$$

$$\dot{v} = \frac{Y_{TOT}}{m+m_y} - u\frac{m+m_x}{m+m_y}\dot{\psi} + \frac{Y_{\dot{\psi}}}{m+m_y}\dot{\psi} \tag{2.54}$$

$$\dot{x} = ucos\psi + u_{tw} + u_{tc} \tag{2.55}$$

$$\dot{y} = usin\psi + v_{tw} + v_{tc} \tag{2.56}$$

where:

- u_{tc} = True current velocity in the *X-axis*
- v_{tc} = True current velocity in the *Y-axis*

From table 2.1 it was defined that ψ is related with $($r$)$ the yaw angular velocity, and it defines the position or orientation of the boat, which means that it set the direction of the course (λ). λ is considered a behavior variable(y_b) in section 2.4. Because it describes the motion of the sailboat, and the seamanship set the direction, this variable is set as a control variable.

This chapter explained how the motion of a sailboat can be determined in 2D with the different types of variables or sources for information. In the next chapter, the environmental variables e are going to be explained and how they interact with the equations of this chapter.

3

The Weather Research and Forecasting Model Application on Sailing

The wind model used in this research is a numerical weather prediction (NWP) known as the Weather Research and Forecasting Model (WRF). This particular model has the ability to forecast weather and serves as a tool for atmospheric research. This chapter explains the characteristics of the common weather forecast provided during and before the Olympic sailing races and discusses the requirements of athletes and coaches. This discussion shows the importance of a customized weather(wind) model and describes its characteristics and limitations.

For the purpose of this project, the model provided is a WRF wind model stored in a NetCDF file, while the measured data from past race events is provided in a tabular layout. The last part of the chapter explains the reference frames and their relation with the sailboat velocity.

The importance of the wind is not only because it is considered the main propulsion source of the sailboat. But also, because it is the wind which induces changes to the seamanship, like direction on the rudder and sail, to balance the kinematic equations of chapter 2, and to reach the maximum velocity estimated by the VPP. In such way, the seamanship uses the wind information to maximize the velocity of the sailboat by adjusting the direction of the sailboat in a similar way the algorithm for path generation uses that information.

3.1. Weather Models Application into Olympic Sailing Races

A wind model forecast is based on a 3-dimensional time-space model, which can be described as a 4-dimensional model. The resolution of this model or granularity is defined by the size of the grid pointsthat describe it. Moreover, the use of weather models during the summer Olympic Games is not new, but neither widely used. In sailing, the main use of them is as an informative source for the organizers. It helps to warrant the safety of the competitors, like in the Para-Olympic events, and reduce delays due to the wind conditions of the competition [25], [26], [27]. Only [28] refers to the NWP model as a tool to develop a strategic plan to course the race for a sailing team.

First competing, athletes and coaches need to know the exact location of the sailing area which is enclosed by a diameter of a magnitude between 0.8 and 1.5 nm (1.482 - 2.778 km) and course diagrams [29]. For them, it is important to identify wind patterns, such as average magnitude, direction, the range of variation, the location of vortexes, and even the time of the day when they occur. This information allows coaches and athletes to develop a strategic plan to course the race by avoiding undesirable conditions or to maximize the use of favorable zones.

Even when the location, dates and times of the races are known, the wind characteristic and details about the area is still not well understood. The time horizons of local forecasts of short term know as *nowcast* for public access is between 1 to 2 hours as a minimum and up to 6 hours, with a spatial resolution between 40 to 100 km [7], [30]. These local predictions use measurements from weather stations or other weather data available around the area of interest and extrapolate these conditions in time. For days or weekly forecast one can use deterministic predictions.

Because sailing Olympic races are set into a grid of 2km with a maximum duration of 1 hour. The public information about wind (weather forecast) does not meet the requirements of the athletes and coaches. As a consequence, customized models have been developed to provide the information required by them. A clear example of this type of models was developed by Giannaros [28], which is defined as a WRF with *ultrahigh resolution wind forecasting.*

An ultrahigh resolution, a detailed level of granularity, means that the spatial variation captured by the model is in the order of hundred meters in the horizontal plane. In this case, the minimum grid size used was about 200 meters on the area of interest and surrounded by different grid sizes up to 25 km. The vertical dimension was defined by 40 unequally spaced elevations or altitudes; while the duration of the forecast stands for 48 hours starting at 0:00 UTC hours each day with a time interval or time step of 30 minutes.

Just considering the area of the course which is placed within a grid of 2 km, a model of ultrahigh resolution has a granularity of 10,200 grid points or 39.5×10^6 grid points. The model developed by Giannaros has even more grid points since the NWP model considers a larger area around the races and other factors which will be explained in the next section. Furthermore, it was calculated using 300 computer cores of a high-performance computing cluster and more than 900,000 core hours [28].

Before [28], only [8] and [31] developed models to include the weather for offshore yacht competitions. Instead of using an NWP model to determine the wind, each study used a different stochastic process to calculate it. The details about the time discretization were not provided. Only [8] acknowledged a short-course competition where the Markov process only applied to the wind direction. The granularity of the model was 4000 grid points with a grid size of 1000 x 1000 meters and a time step of 5 seconds. Other applications of wind models refer to vessels routing where the focus is on fuel costs and other logistics metrics.

The stochastic model used by [31] includes the uncertainty of the wind speed and the direction at each point over the area defined. The wind variables are assumed to be random with a known distribution. The method uses a branching scenario model using the discretized area where the weather evolves. Using the branches the model incorporates the new information while the node represents the actual information. Figure 3.1 shows the scenarios and how the model evolves with the time. where first node is the actual information and where all the branches came out. The assembling of information at each node is a *bundle B(s,t)* where s stands for the scenario at time t. Thus after the first node, the next node represents the possible scenario and the branches its possible outcomes.

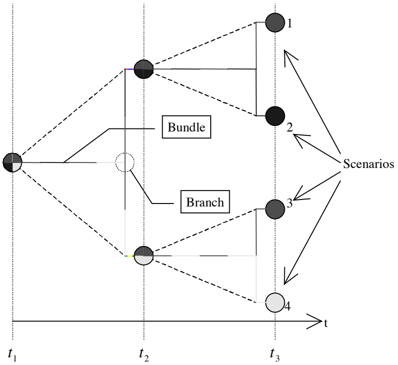


Figure 3.1: Scenario structure with three branches and four scenarios. Identical scenarios treated as one *bundle* [31]

The uncertainty increases by adding the location in the scenario tree, then the node is represented as *(i,t,s)*.

$$t^*(i,j,t,s) = \sum_{s' \in B(s,t)} \frac{p_{s'}}{p_{B(s,t)}} [c_{arc}(i,j,t,s') + f^*(j, t + c_{arc}(i,j,t,s'), s')] \tag{3.1}$$

where $t^*(i,j,t,s) = \sum_{s' \in B(s,t)} p_{s'}$ is the probability of the bundle containing scenario s at time t and $p_{s'}/p_{B(s,t)}$ is the conditional probability of scenario s' at time t given that it is known that one of the scenarios in the bundle containing scenario s will occur, [32].

The other stochastic method used in the study [8] to account for the uncertain weather is a Markov chain. In this study the weather contains a random component related with the time variable. To imitate the observed wind at the sailboat, the wind speed (*W(t)*) and its direction ($\theta(t)$) are discrete independent mean-reverting processes with W (t) reverting to a constant mean wind strength $\overline{W}$, and $\theta(t)$ having a hidden Markov chain structure. The mean to which the wind direction reverts is assumed to follow a discrete-state Markov process,[8] The equations are as follows:

$$W(t) = \overline{W} + \alpha(W(t - \Delta t) - \overline{W}) + \epsilon(t) \tag{3.2}$$

$$\theta(t) = \overline{\theta(t)} + \beta(W(t-\Delta t) - \overline{\theta(t)}) + \delta(t) \tag{3.3}$$

Equation 3.3 is the adopt the Markov chain with a transition matrix, the parameters α and β are constants while $\epsilon(t)$ and $\delta(t)$ are normally distributed errors with zero mean. The states of this Markov chain are a set of discrete angles, and every time step the mean wind direction performs a transition from one of these states to a neighbouring state according to the transition matrix [8]. This model assumes that the sailboat can record the wind measurements.

The WRF model can provide the detailed information required by the sailors and coaches of Olympics Classes. However, it demands time and high-performance computing equipment. Without considering the preliminary input data and validation process it requires to have a feasible and reliable model to work with. At the other hand, the inclusion of weather models for path generation has not been evaluated with respect to the sensibility of the trajectory due to the granularity of the model. The next section will provide the details about the WRF model used for this project as well as its general characteristics.

3.2. Components of the WRF Wind Model

Weather models, like wind or current models, are discretized over the space and time because the data is organized into a four-dimensional grid. The most used computer formats to share this type of models are GRIB(.grib) and NETCDF(.nc). The form in which they organize the information may differ but both formats contain the volume representation of the wind model. A straightforward format sometimes used to share the information is the tabular format with all the variables as headers.

The WRF is a model that combines global atmospheric models with regional measurements to developed high-resolution models. This means that regional models are inserted within the global models, as a result, its range of applications is from meters to many latitude-longitude degrees. The integration allows the identification of small climate variations and other types of phenomena, like precipitations. The incorporation of measured data at a constant rate and at known locations give flexibility to the model [7].

3.2.1. Spatial Representation of the Wind Model

Weather models are covering of a particular region. Because of this, it is important to identify the geometry of the model, to either manipulate or extract properly the variable of interest, like the wind velocity. If the height does not match the CE of the sail, equation 2.1 should be used to determine it.

Figure 3.2 shows the sketches of the most common type of map projections used in atmospheric modeling. The main idea of these projections is that a set of rays coming from the same axis of rotation project points from the sphere into the surface of projection which later can be flattened. The advantage of this procedure is that the angles between curves are preserved and the spatial distortions are the same in all directions. The map projection to be used depends basically on the latitude of interest [7].

The points on the spatial grid of the WRF model are quasi-regular because the grid is defined by the map projection used. Each intersection of the grid is a grid point defined by 4 dimensions ($n(x_i, y_j, z_k, t)$) and it represents the location of a variable value. The

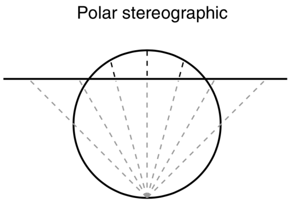


Figure 3.2: Most common map projections used in atmospheric models. Mercator is a cylinder projection commonly used for grids in tropical latitudes, Lambert-conformal is a cone, used for mid-latitude grids and the Polar stereographic uses a plane and is used for high latitude grids. The axis of all 3 always coincide with the Earth's axis. [7].

value of Δx is chosen to have a sufficient number of grid points to adequately represent the smallest meteorological feature of interest.

Under some circumstances, the model uses a nested approach; where Δx is reduced as it gets close to the area of interest. The size of the grid was a crucial concern for [28] since the topography of public databases were replaced by new sets that at least match the size of the stipulated grid.

3.2.2. TIME REPRESENTATION IN THE WIND MODEL

The time dimension on the forecast models is limited by the *Courant number* and in consequence with the spatial resolution required. Modifications on the grid have implications closely related with the computational effort. Since on each time step the space equations at each grid points have to be solved. The measurement data is another element to incorporate and related to time.

Courant number is criterion or condition for convergence and it is used when certain partial differential equations have to be solved numerically. The purpose of this equation is to prevent incorrect results, [7]. *Courant number* is estimated as follow, where Δx is the spatial resolution and Δt is the time step, U represent the speed of a weather wave.

$$\text{Courtant Number} = U\frac{\Delta x}{\Delta t} < 1 \tag{3.4}$$

The computational effort for weather models results from the space granularity. A small time step provides solutions to the equations. While the computational effort is related to the space granularity in a non-linear form. For example, if only one space dimension is doubled (by a factor of 2), this requires four times as many grid point and because the stability criterion based on the Courant number need to be satisfied, the time step will need to halved. As a result the computational effort time will be 8 times more.

The use of measurement data has an impact on the quality of the results because the rate of their incorporation and values adjust the model forecast. Figure 3.3 shows how often the measured data is incorporated into the model. This incorporation determines

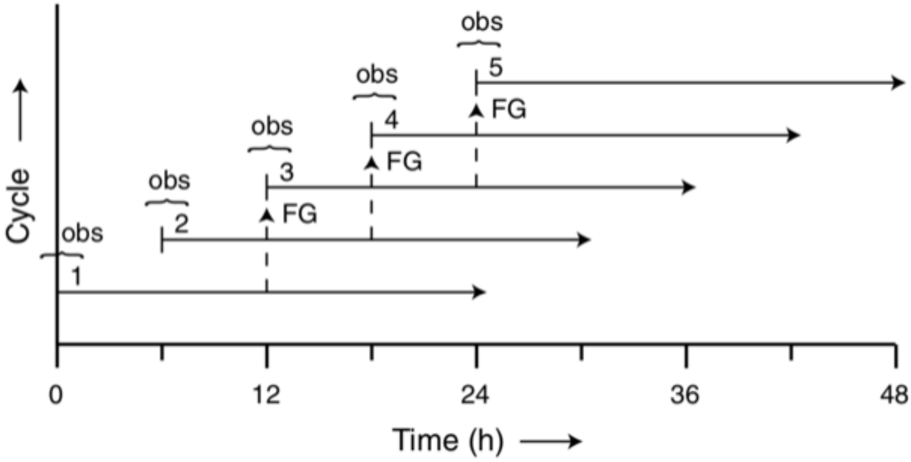


Figure 3.3: Forecast Model with the integration of measured data. During 24 hours, the forecast models are adjusted every 6 hr by the observations. This generates 4 cycles which are assimilated by the model [7].

the rate of the re-initialize and the number of cycles within the model forecast that it has to assimilate the observed values. In this form, the measured data not only affects the time of processing but also the accuracy of the forecast.

3.3. The WRF model for the World Cup Series 2018 at France

The algorithm for the minimal time path for sailing competitions was developed considering the characteristics of the WRF wind model on a date of competitions so wind measurements were available. The measurements were used not only to set-up the model but also to compare their results. Since [28] found that the topography allocation and representation in addition to the initialization of the model have a great impact on the results.

The World Cup Series 2018 Hyéres, France Laser competition was chosen. The area where the races took place was named as Echo and the coordinates of its center were: 43°04.144’N, 006°11.913’ E. The diameter of the area was 1.6 nm (2.96 km). The date was April 24, 2018. The model forecast 24 hours starting at 0:00 Hrs local time and it was provided by professor Sukanta Basu from the Geoscience Department at TU Delft.

The information of the wind model was stored in the NetCDF file which is organized in dimensions, variables and attributes defined as data sets [33]. In this case, 4 coordinates describe the location and time of the wind velocity. The attributes vector refer to the dimensions of each variable, in this case, the velocity’s units were m/s.

The wind speed variables were u and v velocities. Over a rectangle area of coordinates: 41.663°N, 4.752°E and 44.451°N, 7.251°E. The grid size is about 1 km or approximately 0.009°. The coordinates latitude and longitude were stored independently and discretized over a plane of 198×300 elements. Figure 3.4 shows the grid points for the spatial resolution.

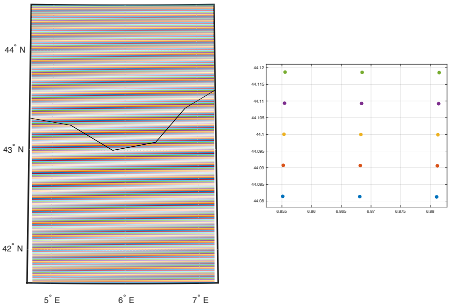


Figure 3.4: Grid Points and close-up of the WRF model for Hyéres, France

The components of the wind velocity were stored on a grid plane of 199×300 for the u and 198×301 for the v velocity. This arrangement is known as grid staggering where the different dependent variables are on different grids so the spatial resolution is increasing while the effects of truncation errors are decreased. The velocity corresponding to each coordinate is the horizontal average velocity calculated as equations 3.5 and 3.6 indicate.

$$u_{i,j_{avg}} = \frac{u_{i,j} + u_{i+1,j}}{2} \tag{3.5}$$

$$v_{i,j_{avg}} = \frac{v_{i,j} + v_{i,j+1}}{2} \tag{3.6}$$

The vertical dimension, height, of the model were discretized over 50 non-uniform steps. The first level corresponds to 7.8 m while the second height is at 25 m. Because the CE is estimated to be at 2.68 m from the water level [34] and the measurements were taken at 10 m above the sea level, equation 2.1 is used to determine the κ value and convert u and v fields to the corresponding height. Finally, V_{tw} and β_{tw} on each grid point can be calculated with equations 3.7 and 3.8, respectively.
Equation 3.8 uses the four-quadrant inverse tangent, (atan2(Y,X)) from $MATLAB^{®}$ to return values over the $[-\pi, \pi]$ interval.

$$V_{tw_{i,j}} = \sqrt{{u_{i,j}}^2 + {v_{i,j}}^2} \tag{3.7}$$

$$\beta_{tw_{i,j}} = atan2d\frac{v_{i,j}}{u_{i,j}} \tag{3.8}$$

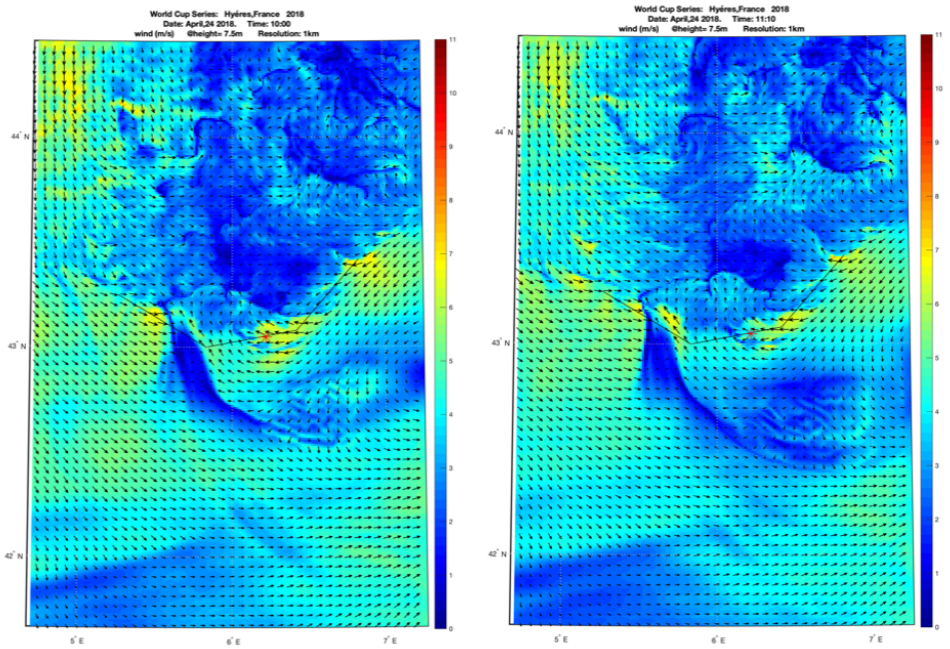
The 24 hrs forecast model was discretized in time steps of 10 minutes or 1/6 hr. This means that there are 145 steps on the time dimension. Figure 3.5 shows the wind model provided at 2 different hours, the time difference is only 1 hour and 10 minutes. The red cross is the center of laser course area named Echo and the black arrows indicate the direction of the wind. The figure also shows the relevance on the scale or granularity. On one side public forecast weather are larger than the one showed here. In the other hand, the interested region is too small even when the granularity was increased significantly.

In summary, the wind model provided is a 4D(*(x,y,z,t)*) grid model of 198 × 301 × 50 × 145 grid points with a spatial resolution (Δx) of 1 km and time step (Δt) of 10 minutes(600 seconds). This kind of granularity is defined as high resolutionThis resolution capture small variations over the space but not as smaller as the model by [28]. Because of this, a smaller resolution refers to ultrahigh spatial resolution.

3.4. Reference Frames the relation between wind and sailboat velocities

The wind velocity and direction determines the velocity of the sailboat. However, the wind velocity perceived by the seamanship when the sailboat moves is not the same as the velocity of the sailboat. This perceived velocity is known as the *apparent wind* (V_{aw}) and it not only includes the velocity and direction of the wind but also the velocity and direction of the current (V_{tc} and β_{tc}). The V_{aw} and β_{aw} are not new concepts, they have been introduced in section 2.3.1 and in equation 2.2.

The V_{aw} is modified by the current in a similar form as in equation 2.2. From this


World Cup Series: Hyéres,France 2018
Date: April,24 2018. Time: 10:00
wind (m/s) @height= 7.5m Resolution: 1km
44° N
43° N
42° N
5° E
6° E
7° E
0
1
2
3
4
5
6
7
8
9
10
11
World Cup Series: Hyéres,France 2018
Date: April,24 2018. Time: 11:10
wind (m/s) @height= 7.5m Resolution: 1km
44° N
43° N
42° N
5° E
6° E
7° E

(a) Wind model forecast at 10:00 Hrs

(b) Wind model forecast at 11:10 Hrs

Figure 3.5: Wind Model for the World Cup Series 2018 at Hyéres, France. The red asterisk indicated the center of the Echo area for the Laser Course. The area of the model is defined by the corner coordinates at 41.663°N, 4.752°E and 7.251°N, 44.451°E.

equation the current vector is subtracted as shown in equation 3.9 and this velocity is defined as $V_{a_{w,c}}$. The vector expression shows how different directions influence the results even when the magnitude is the same. On the other hand, if the boat is moving in the same direction as the wind then the apparent velocity increases [6],[32].

$$\overrightarrow{V_{a_{w,c}}} = \overrightarrow{V_{tw}} - \overrightarrow{V_{tc}} - \overrightarrow{V_{boat}} \tag{3.9}$$

However, the previous equation is not the only modification to consider related to wind. The wind model only applies over a specific region on the Earth-fixed frame, while the sailboat frame moves along it and is located as indicated by section 2.5. It was assumed that the area and the trajectory of the sailboat can be expressed as $[X,Y]$ coordinates. This is possible for Olympic Sailing Races because the course area is relative small compared with the total surface of the Earth. Thus to convert the latitude-longitude to $[X, Y]$ coordinates, the $MATLAB^{®}$ function *deg2utm(Lat, Lon)* can be used [32].

For the purposes of this research, it is assumed that the Earth circumference is about 40003.2 km. In addition, the nautical mile is one minute (1\60°) of arc on the surface of this sphere, thus the nautical miles is 1852 m.

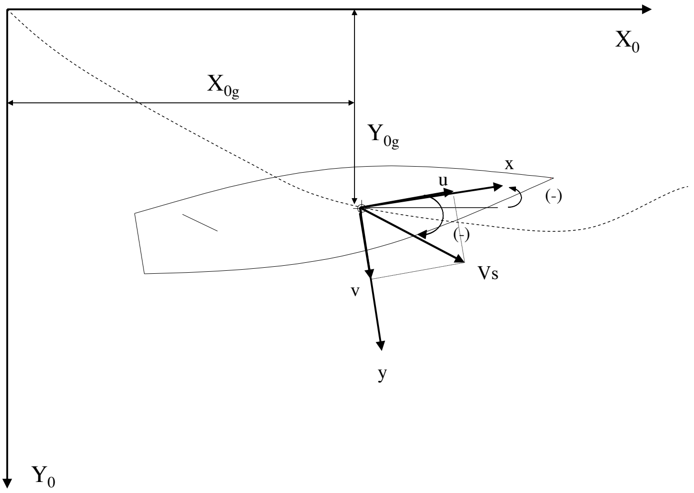


Figure 3.6: Sailboat reference coordinate system and Earth-fixed reference frame [22].

Since the sailboat moves through the horizontal plane of the wind model, another transformation has to be applied in order to describe the motion and forces of the sailboat. Figure 3.6 shows how u is always aligned with the sailboat mid-plane, therefore to its heading direction (-Ψ). The change in orientation with respect to the same reference frame is done with the rotation matrix 3.11. This transformation is shown in equation 3.10 and it expresses the velocity from the Earth-fixed frame to the sailboat-fixed frame V_{tw}^{b} [21], [19], [15], [22].

$$V_{tw}^{b} = \mathscr{R} \cdot V_{tw} = \mathscr{R} \begin{bmatrix} u \\ v \end{bmatrix} \tag{3.10}$$

$$\mathscr{R}(-\Psi) = \begin{bmatrix} cos\Psi & sin\Psi \\ -sin\Psi & cos\Psi \end{bmatrix} \tag{3.11}$$

If the V_{tc} is included in the model without waving effects the V_{tw} has to be modified before the transformation of equation 3.10. This is expressed in equation 3.12 [32]. However, more equations have to be redefined because of this inclusion. The equations modified are

$$\overrightarrow{V_{tw,c}} = \overrightarrow{V_{tw}} - \overrightarrow{V_{tc}} \tag{3.12}$$

$$V^b_{tw,c} = \mathscr{R} \cdot V_{tw,c} \tag{3.13}$$

$$\overrightarrow{V^b_{aw,c}} = \overrightarrow{V^b_{tw,c}} - \overrightarrow{V_{boat}} \tag{3.14}$$

$$\beta_{aw,c} = atan2d\frac{v^b_{aw,c}}{u^b_{aw,c}} \tag{3.15}$$

The transformation of the terms from the sailboat-fixed to the Earth-fixed coordinate system and V_{tw} using $\mathscr{R}$ is not necessary on the equations from 2.5 since they were already included in their development [22].

This section explained the WRF wind model used for the optimization of trajectories on Olympic Sailing Races. The characteristics of this model define some of the parameters of the Path Algorithm, therefore for the Minimal Time trajectory. The scale of the Olympic Sailing Courses is small compared with the coverage of wind models. The impact of this coverage hasn't been evaluated and the use of customized models is very limited. Even when the algorithm to be developed does not include any model for currents, and it is assumed that current velocity V_{tc} is constant some modifications have to be introduced.

4

Time Optimization Algorithm for Laser Sailing Paths.

The optimization problem for the minimal path is a problem frequently related to logistics and operations research. This problem is usually associated with the *Euclidean* shortest path and solved either with networking optimization or dynamic programming (*DP*). Although for Olympic sailing classes the optimal path is not related to distance but with time instead.

Because of this, the research of this type of problem in sports is minimal and it is mainly focused on yacht competitions. Whereas the laser class is the smallest and one of the most used sailboats in Olympic Classes. The objective of this section is not only to explain in detail the methods used but also their elements and how the algorithm for the Laser Olympic class apply these elements to optimze the time-path.

Many techniques used on DP and networking optimization can be used in combination with the VMG criterion to develop an optimization algorithm for laser races. The target's location and the geometrical domain are the features that allow the use of both criteria, thus the optimal solution is found inside the polygon [35].

The first part of this chapter explains how chapter 2 and chapter 3 are integrated to develop the algorithm to optimize the path of laser boats and obtain a minimal time path. But because most of the formulas on section 2.3.2 are generic, the adjustments required to describe the laser class are explained in the second part of this chapter, followed by the explanation about the algorithm, in detail such as the objective function, the constraints, and its validation. The validation of the algorithm is at the end of the chapter, and also the information about the parameters defined by the user provided before the initialization of the algorithm.

4.1. WEATHER ROUTING MODELS AND PATH ALGORITHMS FOR SAILBOATS

Uncertain weather is a typical condition that not only yacht competitions have to manage but also maritime transportation. The direction-dependency of the vessel's velocity is seen as a series of regions where the flow's velocity is designed as uniform. This is characterized as an anisotropic medium condition, and path algorithm problems with this characteristic have been solved with different methods [36].

The most used methods identified with vessels or yachts and anisotropic medium come from operations research and logistics. For example, from operations research, these methods are dynamic programming (DP), direct and indirect methods while the networking method is associated with logistics. [37],[35]. In the networking method, the order in which the locations can be reached is not as important as time.

The formulation of the problem is to focus on the time and not in the length of the trajectory. However, it is the velocity of the sailboat the one that determines the direction of the sailboat and considers the wind properties. Because of this, equation 4.1 defines the time of the path to displace a sailboat between 2 points [38]. This is the easiest formulation of the problem but it shows how this problem differs from the shortest path approach.

$$T_{AB} = \int_{t_A}^{t_B} dt = \int_{l_o}^{l_f} \frac{dl}{v} \tag{4.1}$$

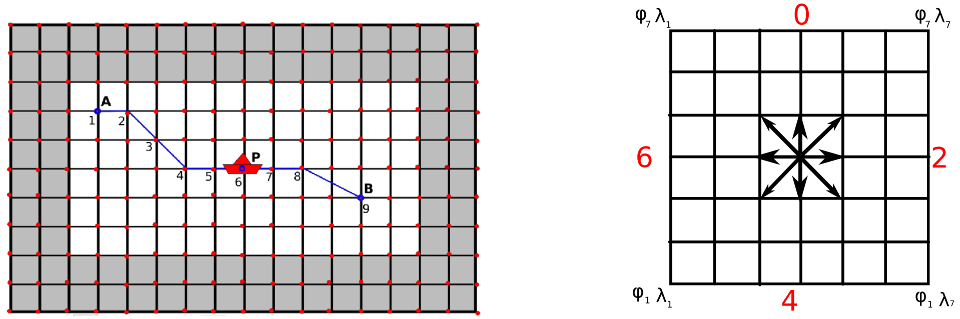
4.1.1. DYNAMIC PROGRAMMING AND DIRECTION-DEPENDENCE FOR PATH ALGORITHMS

DP is a widely used technique to solve the optimal path problem for yachts and maritime transportation. The reason is that it breaks the main problem into multiple stages that are all connected. In this case, to move from point A to point B the trajectory is composed of more than two points. Meaning that the main trajectory is assembled by multiple stages or smaller trajectories continuously coupled until it arrives at its destination.

The fact that one stage depends on the previous one, allows the recognition of the state of the variables implicated in that solution. This set of state variables is used to optimize the trajectory by iterating each stage until the optimal solution by stage is found. Thus, the next stage always starts from the optimal solution given by the previous one [8].

In yacht competitions and maritime transportation, the trajectory is not only defined by the location of the points but also within the area they stand. Hence the algorithm first have to define the area and later discretize it. The discretization serves not only to describe the variables that depends on its location and time, like wind and current but also to apply DP. For this reason, the discretized area is composed by nodes and the line that connects 2 nodes is an arc (c_{arc}), figure 4.1 shows an example of this.

This process discretize the area using a grid, here the maximum velocity of the sailboat drives the heading decision. This method is similar to the VMG criterion because the same distance from the starting point can be reached at different times. For the heading-angle decision the discretization use intervals of size $\Delta\Psi$, clockwise and counterclockwise, as a result, multiple sub-routes are generated, like in figure 4.1b. The fig-



(a) Sailing Area Discretization for a Sailing Path from point A to B [39].

(b) Heading Directions at a node, 8 angle discretizations [39].

Figure 4.1: Area discretized with a sailing path and the Heading Angle discretization

ure indicates 8 heading-angles at one node, this means the interval is about 45º. If the discretization takes into account the symmetry of the VPP the shape developed by the nodes and the subroutes is a diamond, as in figure 4.2, [40] [39].

The number of stages(n_{stages}) along with the straight sailing distance (L), between point P_0 and P_n, determines the shortest distance between ($L \setminus n_{stages}$). In this case, n_{stages} has any value bigger than 1, this parameter defines the number of attacks that the seamanship can perform. However, the bigger the number the longer the time of the computational effort to determine the trajectory of the optimal time path [40].

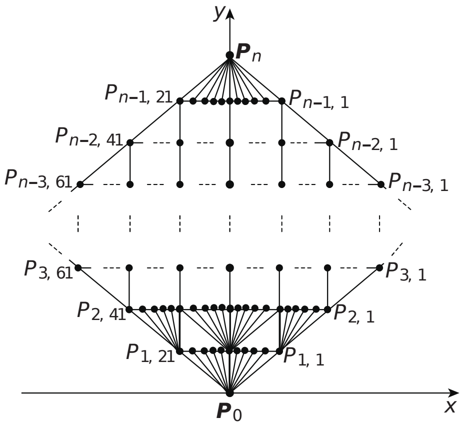


Figure 4.2: Diamond Shape with the Sub-routes for a heading-angle discretization to move from P_0 to P_n [40].

Figure 4.2 is an example of how the heading decision works. In this particular case, all the sub-routes from P_0 to P_n are contained into a diamond shape. The notation of the nodes indicates the stage and the node-number per stage, $Pn_{stage}, i_{sub-route}$, these numbers result from the size of $\Delta\Psi$. The nodes over the edges are the half of the nodes compared with the nodes at the center. The number of clear stages represented in the figure is 6 and on each node, the maximum number of sub-routes is 21 and the minimum is 10. If the algorithm integrates the concept of the *no-go-zone* from section 2.4 the number of sub-routes decreases if $\Delta\Psi$ is smaller than the no-go-zone angle.

To incorporate the time at the node, for example an arc to connect the node i and j the expression is $c_{arc}(i,j,t(i))$, where c_{arc} is the expression to define that two nodes are connected, its value depends on the nodes that connect (i and j) and the time (t). Thus, it provides the time to connect node i to node j. In this expression the arc also depends on time, represented by (t). At the starting point(t_A), the wind ($\mathbf{w}(i,j,t)$) and the current($\mathbf{c}(i,j,t)$) characteristics are known, and by using DP the equation to estimate the minimal time path is equation 4.2 [39],[31]. The optimal heading is finding when the time to reach the next stage is the minimal. Following this, it is possible that multiple nodes arrive at the same time to the next stage. For this reason, the algorithm have to store the sub-routes to continue the process and compared the results(time) at the end, once the final point has been reached.

$$f^*(i,t) = \begin{cases} 0, \\ \\ \min\limits_{j\in\Gamma_i}\left[c_{arc}(i,j,t) + f^*\Big(j, t + c_{arc}(i,j,t)\Big)\right] \end{cases} \tag{4.2}$$

where

$$j^*(i,t) = arg\min_{j\in\Gamma_i}\left[c_{arc}(i,j,t) + f^*\Big(j, t + c_{arc}(i,j,t)\Big)\right], \quad i \neq n_{finish} \tag{4.3}$$

This equation explains how the time is minimized at each node until the final destination, point B, is reached. $f^*(i,t)$ is the time at node (i,t) which is optimal time resulting from the previous stages. $j^*(i,t)$ is the next node from i on the optimal path, while Γ_i is the set of subsequent nodes of i. The minimal time of equation 4.2 to move from node i at time t when the next node is j is given by:

$$c_{arc}(i,j,t) + f^*\Big(j, t + c_{arc}(i,j,t)\Big)$$

Figure 4.3 is the graphical explanation of the equation that determines the minimal time from node 1 to 5, starting at time 1 ($n(1,1)$). In this example, Node 5 (n_5) can be reached by 3 different paths, or in other words, there are 3 alternative nodes before reacing n_5. The nodes 2, 3 and 4 can be reached at different times, the minimum time between them is given by n_4 with a time of 2 ($n(4,2)$). The next arc is formed from n_4 to n_5 and because the starting time of n_4 is the minimal/optimal time for the first stage, the final time is then the minimum time path from n_1 to n_5.

This state-space algorithm for the shortest path includes explicitly the time dimension [32]. Besides, the heading-direction approach sometimes uses additional factors over directions to maximize the speed.

.

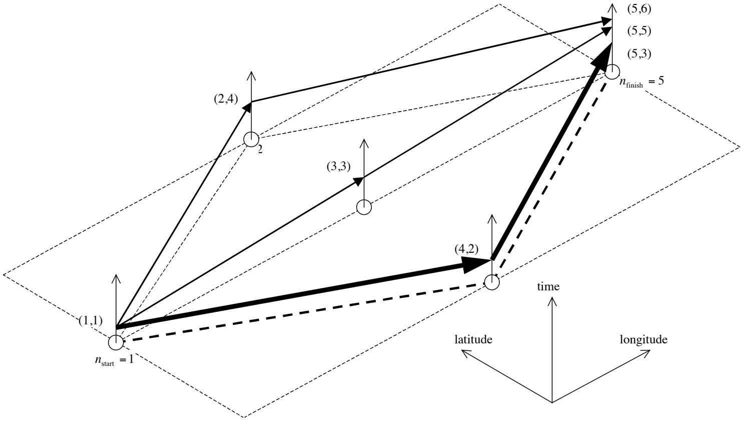


Figure 4.3: Minimal Time Path from node 1 at time 1 (*n(i,t)=n(1,1)*) to node 5 using DP. The minimal time path to move from node 1 to 5 has gone through node 4 and the path is indicated by the thicker line [31].

4.1.2. PATH ALGORITHM USING ISOCHRONES

A common solution with visual information obtained with DP in weather routing is the isochrones. The isochrones lines compose a map where each line shows the maximum distance a sailboat can reach during a certain time. Moreover, this map can be seen as the visual representation of the VPP over different anisotropic media when the weather variation is minimal [32].

A similar approach relates this type of solution with geometrical optics more specific with wavefronts. This analogy is because the speed of the wavefront depends on the refractive index of the material [38]. Figure 4.4 shows how *qtVlm* software uses isochrones to determine the optimal path for a generic yacht.

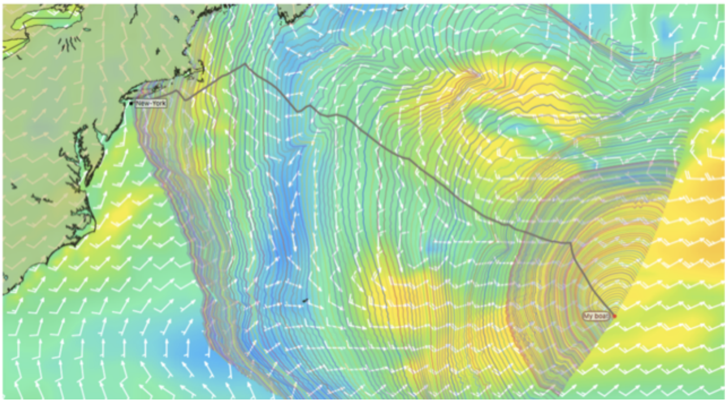


Figure 4.4: Optimal Path Solution from qtVLm software using isochrones for a generic Yacht from a random location over the Atlantic to New York. The optimal path is the gray thick line, the wind direction is indicated by the white arrows [38].

The visual information presented by most of these algorithms has been employed by various weather routing software. Despite this, the use of them in sports and in Olympic Sailing Classes is limited. This is because they are designed for long-distance yachts races or for maritime-logistics purposes. In both cases, the weather model is assumed to be homogeneous in space and time and frequently loaded from public sources. Meaning that the time step and grid size is much bigger than the required for Olympic Sailing races. Furthermore, when it is used for maritime-logistics purposes the vessels are equipped with communication and other measuring systems that account for weather variations.

Despite the advantages of the isochrones techniques and its usage on large course races, the number of application for short courses is limited. The reason is because of the weather is assumed to be perfectly known and even when the VPP is assumed symmetric the algorithm doesn't explain or shows the limitations of choosing the symmetric path not even as an alternative for some space-time-intervals.

Summing up, DP is a flexible method used to find the optimal sailing path and widely used for weather routing. The main characteristics of the previous techniques that have been considered for the development of the laser path algorithm are explained afterward. Before starting any sailing path, first, it is important to define the area within any path might take place. This is not only because of the wind or current model but because inside that area, a set of stages have also to be defined. This area for the sailing path has to cover effectively all the alternatives so the minimal time path can be found.

After the area is defined it has to be discretized, the grid approach is commonly used. By using the grid it is easy to locate the node for either the node or the arc and to find the value for both wind and current. The discretization is directed related to the granularity of the problem and at the same time with the computational effort.

The next characteristics to define is the number of stages is a free number that must be taken with care. The number of stages in combination with the number of variables define the state-space vector. This vector has to be estimated and later on compared as many arcs or nodes were determined. As a consequence of the size of the vector the computational effort can be affected and at this point, the number of operations could grow exponentially.

4.2. Adapting the Yacht Model to the Laser Class

Most of the research either for path planning or for the physic model for sailboats are associated with the yacht or with bigger boats such as sailing vessels. The figure shows the differences in heights for the mast only, where the AC45 yacht race has a height of 25.5 m while the Laser Olympic class mast's height is 6.1 m. Equations on section 2.5 are generic and applicable to any kind of sailboat. Hence, to represent properly the laser class some modifications have to made before to properly represent its motion.

The sailboat of the laser class is the dinghy one of the smallest boats propelled by wind, with a maximum weight of 59 kg [3]. Therefore, its balance on the *heave axis* (*Z-axis*), between the heel angle (Θ) and righting moment (M_R) is determined by the ability of the crew to adjust its posture over each side of the boat. The posture of the crew in respect to the boat is determined by the wind speed and direction mainly. So it can be said that the crew's posture is a response to the weather variations [2].

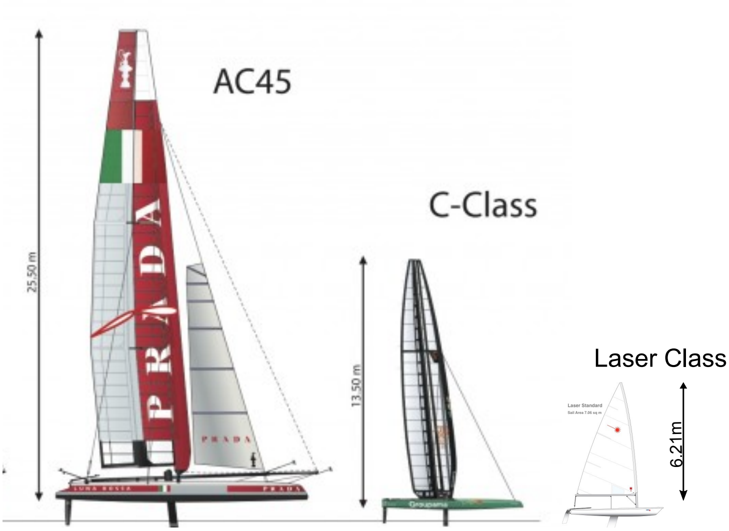


Figure 4.5: Mast Height Comparison between two yachts models AC45 and C-Class versus the Laser Olympic Class, side view. [41].

Even when these assumptions are implicit to keep the sailboat analyses in 2-dimensions and developed the VPP there is some discussion about it [11],[13]. The discussions are associated not only with the posture but also with the impact of the crew's mass (m_c). Since for Olympics Sailing Races, it could represent more than 50% of the total mass of the sailboat (m)[18]. These without considering the rate of change of these postures adjustments and the assumption that its centroid (m_c) is located in the center of gravity (CG) of the sailboat.

A deeper analysis and comparison between the adaptations on the standard VPP were addressed about the coefficients and forces related to the sails. The comparison of the different coefficients values include the data taken from the Offshore Rating Congress (2013) and the observations made in other works [18], [17], [42], [43]. Furthermore, these modifications correspond to the addition of the *reef* and *flat* coefficients of the sail. Particularly, they account for the fact that dinghy's sail does not reef against strong winds. As a consequence of this, the drag and lift coefficients must be adjusted during the upwind and downwind course [17].

The adaptations directly associated with the aerodynamic forces, as a result of the *flat*, *twist* coefficients and the *spill*(s) variable are showed next. These coefficients model the behavior of the sails and the athlete under strong wind conditions and it includes not only the upwind conditions but also the downwind course [18]. Therefore, equations 2.17 and 2.18 are modified to account the required adjustments.

Then s modifies equation 2.18 and is shown in equation 4.4 where C_{twist} has a value of *8.0* and t which is the *twist variable* has a range value of [0,1], C_{Dv} and C_{Lmax} is obtained by interpolation from tabulated values and it depends on β_{aw} while C_{Ds} has a value of *0.005* and AR_E is based on the rig geometry and calculated according to it [18].

Equation 2.17 is replaced by 4.5, the values taken were the smallest with a reduction of the area of 20 % to consider shielding from the cockpit [18].

$$C_t^* = C_{Dv}(\beta_{aw} - s) + C_{Dp} + f^2 \cdot C_{Lmax}^2 \cdot (\beta_{aw} - s) \cdot \left(\frac{1 + C_{twist} \cdot t^2}{\pi AR_E} + C_{Ds} \right) \tag{4.4}$$

$$C_d^* = 1.075 \qquad \text{for the frontal area} \tag{4.5a}$$

$$C_d^* = 0.954 \qquad \text{for sideways area} \tag{4.5b}$$

The literature regarding VPP for laser boat classes is limited and it is mainly accounted in [18], the comparisons made here are only valid for wind speed between 4 and 16 knots, while races are performed up to 25 knots. This range has to be considered in the development of the algorithm to find the optimal path since it is the velocity of the wind that determines the direction that should be taken in order to maximize the VMG criterion.

4.3. THE MINIMAL TIME PATH OPTIMIZATION ALGORITHM

At this point, all the elements required to define the algorithm for the sailing path have been described. In this section, those elements are going to be implemented in a similar order in which they were described. First, the parameters of the Laser are going to be described so its VPP can be estimated, then the area for the sailing race has to be defined and finally how the optimization was setting up and validated.

4.3.1. THE LASER OLYMPIC CLASS

As mentioned before the Laser sailboat is the smallest class of the Olympics Sailing Classes and it is shown in the figure. The dimensions and parameters that describe the laser are regulated by the International Laser Class Association (ILCA) and they are shown next in addition to some coefficients previously defined. The figure shows a side view of the Laser Standard and Laser Radial which are Olympic Classes.

m_{boat}= 59 [kg] Mass of the sailboat

Loa = 4.2 [m] Overall Lenght

Beam = 1.37 [m] Beam width

T_c = 0.787 [m] Draft of canoe body

$A_{s,radial}$ = 5.76 [m^2] Sail Area for Laser Radial (women category)

A_{sail} = 7.06 [m^2] Sail Area

A_{keel} = 0.23 [m^2] Keel Surface Area

A_{rudder} = 0.11 [m^2] Rudder Surface Area

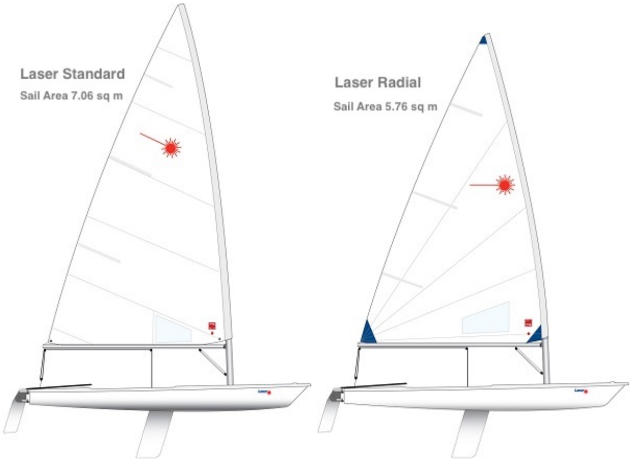


Figure 4.6: Laser Olympic Classes. Laser standard refers to men category while Laser Radial is for women. The difference between them is only the size of the sail. [44]

$C_{D,hull} = 0.02$ [-] Hull Drag Area Coefficient

The equations of section 2.5, use the added mass over each axis which according to [22] can be determined by equations 4.6 and 4.7. At this point, the m_c is required and for the purposes of this research, the maximum value suggested by [10] is used.

$$m_x = 2\frac{T_c}{Loa}m_T = 2\frac{T_c}{Loa}(m_{boat} + m_c) = 2\frac{0.787}{4.2}(59+70) = 48.34 \text{ kg} \tag{4.6}$$

$$m_y = 0.25 \cdot m_T = 0.25 \cdot (59+70) = 32.25 \text{ kg} \tag{4.7}$$

Most of the coefficients used on the equations of section 2.5 can be found on tables, however, they can be approximated by using sine and cosine functions [45],[46]. These coefficient approximations are referred to the keel ($C_{i,keel}$) and rudder ($C_{i,rudder}$) only and they are shown next, the sub-index L is for the *lift* while D is for the *drag*.

$$C_{L,keel} = 0.615sin(2\beta_{k_a}) + 0.025 \tag{4.8a}$$

$$C_{D,keel} = -0.55cos(2\beta_{k_a}) + 0.55 \tag{4.8b}$$

$$C_{L,rudder} = 0.6175sin(2\beta_{r_a}) + 0.0325 \tag{4.9a}$$

$$C_{D,rudder} = -0.55cos(2\beta_{r_a}) + 0.55 \tag{4.9b}$$

Due to these approximations equations 2.25, 2.26 and 2.27 are modified and facilitate the separation of the *lift* and *drag* forces to use them. The angle of attack (α_a)is related to the β_{aw}, therefore it is associated with the trim angle of the sail (δ_s). The keel is a fixed element and it is assumed rigid then β_{k_a} is the same as the β_{ac}.

$$\beta_{ac} = atan2\frac{v}{u} \tag{4.10}$$

$$V_{ac} = \sqrt{u^2 + v^2} = V_{boat} \tag{4.11}$$

$$F_{L,keel} = \frac{1}{2} \cdot \rho_w V_{ac}^2 A_{keel} C_{L,keel}(\beta_{ac}) \tag{4.12}$$

$$F_{D,keel} = \frac{1}{2} \cdot \rho_w V_{ac}^2 A_{keel} C_{D,keel}(\beta_{ac}) \tag{4.13}$$

$$\beta_{r_a} = \beta_{ac} - \delta_r \tag{4.14}$$

$$F_{L,rudder} = \frac{1}{2} \cdot \rho_w V_{ac}^2 A_{rudder} C_{L,rudder}(\beta_{r_a}) \tag{4.15}$$

$$F_{D,rudder} = \frac{1}{2} \cdot \rho_w V_{ac}^2 A_{rudder} C_{D,rudder}(\beta_{r_a}) \tag{4.16}$$

$$\alpha_a = \beta_{aw} - \delta_{sail} \tag{4.17}$$

The motion of the sailboat is dominated by the wind and this research is focused on the effects of the wind. As a result, the β_{i_a} of the rudder and keel are treated as one component and substituted by α_a [45]. The implication of this substitution has two effects, first on the R_{hull} and second on the lateral force. The coefficient related to the R_{hull} has to account for, thus it has to increase to *0.025*. **The lateral forces are assumed to be in equilibrium all the time, as a result, the v velocity known as drift speed is neglected** hence equation 2.54 is omitted from the equations of motion for the Laser Class [45].

Another modification is identified with the equation 2.53 which has to be modified to replace the hydrodynamic derivative expression ($X_{V\psi}$) by a damping expression showed on the equation . The damping expression is related to a constant value (*Cnst*) of 0.3 according to [45].

$$\dot{u} = \frac{X_{TOT}}{m + m_x} - Cnst \cdot \dot{\psi}^2 \tag{4.18}$$

To recapitulate all the previous changes, the equations of motion that describe the motion for the Laser Olympic Class are:

$$X_U = \frac{1}{2}\rho_w V_{boat}^2 \cdot C_{D,hull} = \frac{1}{2}\rho_w \cdot V_{ac}^2 \cdot 0.025 \tag{4.19}$$

$$X_{sail} = F_{L,s} sin\beta_{aw} - F_{D,s} cos\beta_{aw} \tag{4.20}$$

$$X_{current} = m \cdot V_{boat}\dot{\psi} = m \cdot V_{tc}^b \cdot \dot{\psi} \tag{4.21}$$

$$X_{TOT} = X_U + X_{sail} + X_{current} \tag{4.22}$$

where:

$$F_{L,s} = \frac{1}{2}\rho_a V_{aw}^2 \cdot A_s C_t^* \tag{4.23}$$

$$F_{D,s} = \frac{1}{2}\rho_a V_{aw}^2 \cdot A_s C_d^* \tag{4.24}$$

Finally, the equations that describe the motion are:

$$\dot{x} = u \cdot cos\phi + u_{tw} + u_{tc} \tag{4.25}$$

$$\dot{y} = u \cdot sin\phi + v_{tw} + v_{tc} \tag{4.26}$$

$$\dot{u} = \frac{X_{TOT}}{m + m_x} - 0.3\dot{\psi}^2 \tag{4.27}$$

The optimization of the time-path problem for Laser is described as follow:

$$\text{maximize:} \quad u(u, \psi, \delta_s, V_{tw}) \tag{4.28}$$

$$\text{subject to:} \quad \dot{u} = 0 \tag{4.29}$$

$$\dot{\psi} = 0 \tag{4.30}$$

These modified equations serve only for the laser Olympic class and with them, the VPP can be estimated. Even so, at some wind velocities the equation 4.4 could be less accurate, especially at V_{tw} close to 20 kn and above it. Since equations 4.4 and 4.5 work fine when the V_{tw} is close to 10 kn [18], which is at the lower range of the V_{tw} range defined by [10].

4.3.2. VPP FOR THE LASER OLYMPIC CLASS

The optimization of the minimal time path as showed in equation 4.1 requires to determine first the velocity which in this case $V_{aw}(\psi)$. $V_{aw}(\psi)$ is related to the wind speed and direction and considering the computational effort and the fact that the wind model is space and time discretized. The VPP can be estimated first since the wind speed range rule ([4,25]kn) conditions the laser races. To estimate the VPP not only the β_{tw} but also the V_{boat} has to be discretized with small steps values.

Using the previous equations and replaced in equations from 2.49 to 2.56 the VPP can be obtained using an optimization method. The systems of equations can be solved over the angle range of [0,180 °] using the built-in function of *fmincon* from $MATLAB^{®}$[45]. The problem is defined as maximization of the V_{boat} which means that all the forces are in equilibrium. However, any optimization problem has to be defined in terms of the minimum value, then the problem is defined as:

$$\text{maximize:} \quad u(u, \psi, \delta_s, V_{tw}) \tag{4.31}$$

$$\text{subject to:} \quad \dot{u} = 0 \tag{4.32}$$

$$\dot{\psi} = 0 \tag{4.33}$$

Equation 4.31 shows that u is the boat's velocity and it depends on itself which means that the system is not linear. Moreover, its influence on β_{tw} V_{aw} determines the sail's forces. The control variables this problem are $\dot{\psi}$ and δ_s, this last could have a value over the range of [0,180 °] but it should not cancel Ψ, during motion.

Figure 4.7 shows the full VPP of the Laser Olympic Class for a wind of 8 kn. In this graphic, the direction of the wind is zero degrees from the North. The angle's interval is 10° and the V_{boat} is indicated by the diameter of the circles, in this case, it has a range value of [0,3.3] m\s. In the figure, the *no-go-zone* is in the angle range of [0, 40] degrees

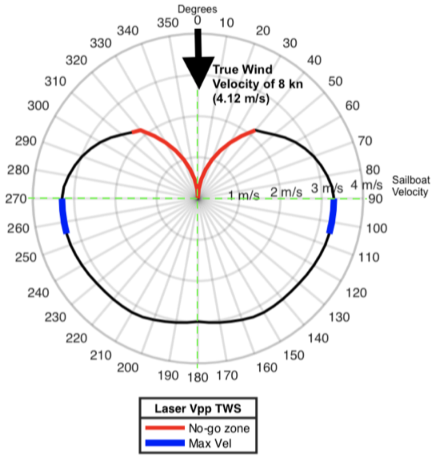


Figure 4.7: Full VPP for the Laser Class at 8 kn V_{tw} coming at 0° from the North, the angles were discretized with an interval of 10°

and [320, 360] degrees while the maximum velocity of the boat can be reach in the angle range of [90, 115] degrees and [255, 270] degrees.

The V_{boat} range taking out the *no-go-zone* is [2,3.3] m\s, the VPP shape over the range of [90,270] degrees does no vary too much. The shape is close to a semi-circle, this means that under a downwind condition a straight trajectory could be more efficient if the tack loss time is bigger than the velocity change ratio while the V_{tw} remains are constant.

An alternative method to estimate half of the VPP uses wind measurements for the speed and its direction. These measurements, however, do not have a constant interval, so to predict any other wind speed and its direction, the measured data is interpolated [8],[31]. For the purpose of this project, some wind measurements for wind speed were provided by $InnoSportLab^{®}$, The Hague.

The interpolation used for the missing points was the Piecewise Cubic Hermite Interpolating Polynomial (PCHIP) from $MATLAB^{®}$. The use of this data not only serves as validation for the VPP calculations but also as a data source. This because of many of the research regarding VPP for the Laser Olympic Class only cover a wind speed range of [9,12] kn [18] and the measurements provided are out of this range and they are shown in figure 4.8.

Figure 4.8 indicates the measurements taken with a black asterisk, the rest of the points, therefore the line is the result of the PCHIP interpolation. The wind measured is referred to as *TWS* and it is assumed to come from the North, top of the graph. The VPP is assumed to be symmetric and due to this, the measurements were only taken in the range of [0,180] degrees.

The measured data not only serves to develop the VPP but also to validate some of the assumptions previously made. Now that the VPP was determined the formulation of the minimal time path for the Laser Sailing Class can be described.

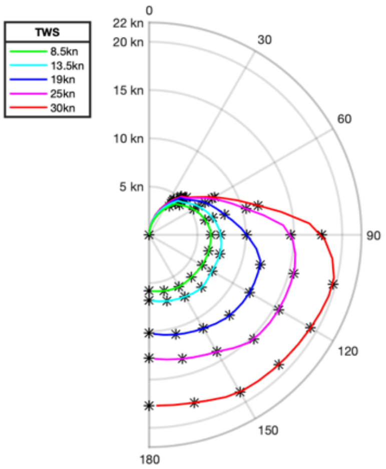


Figure 4.8: Vpp developed with measurements provided by *InnoSportLab*®, The Hague. The measurements are indicated with a black asterisk. The results of the interpolation vary according to the wind velocity (TWS)

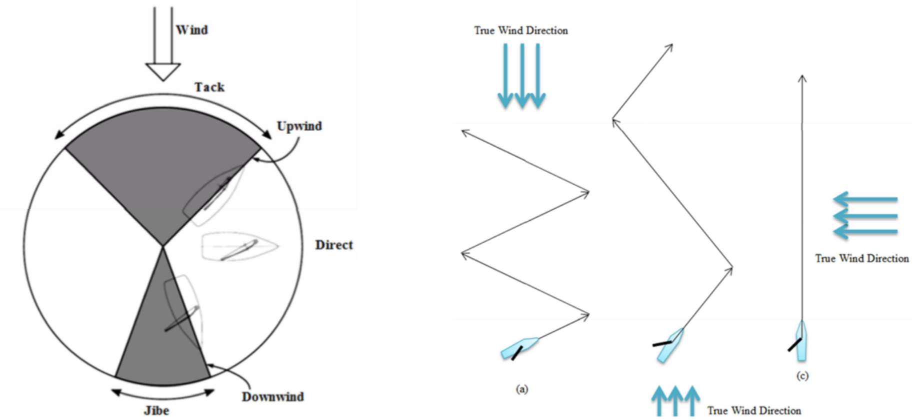
4.3.3. The Objective Function: The Minimal Time Path

At this point, all the elements required to develop the algorithm have been explained. In this section, their implementation is going to take place. The objective is to find the path with the minimum time, regardless of the type of sailing course. Figure 4.9, shows the three types of courses with its main maneuvers and an angle range where they take place. The angle range is not specified since it depends mainly on the VPP (according to the sailboat) and sailor preferences. For example, under a downwind condition, many sailors prefer to follow a straight line rather than a zig-zag pattern.

The combination of at least two of these sail-modes determines a race for Olympic Sailing Competitions. The *Leg* is the section of the race defined by at least two buoys. Each leg is subject to a particular sail-mode, this means that the range of directions respect to the wind are different and the minimal time for the race is given by the sum of the minimal time on each leg. The direction of the leg respect to the wind determines the maximum velocity that can be reach, moreover the number of maneuvers or changes in direction to perform.

The identification of the sail-mode over the leg determines for example if a straight line is possible or not. This helps the algorithm to reduce the number of directions to evaluate and focus on the ones that improves the velocity. This optimization problem is defined as a multi-phase problem, where each *leg* is the phase of the problem. It is a multi-phase problem not because all the phases are connected but because on each phase different sets of constraints are implemented.

Each leg of the course to race is defined by the index *i* and it starts at 1. The number of stages or states of the variables is defined by the index *n* and it can be described as an intermediate set of points inside each *i leg* where the time and distance are evaluated. This means that if there are two legs in total and the number of stages is 5, therefore there



(a) The 3 main modes to sail respect to the wind direction.

(b) Types of Maneuvers for sailing according to the boat and wind direction.

Figure 4.9: Type of courses and its Maneuvers [15]

are twelve stages in total ($n \times i = (5+1) \times 2 = 12$).

The approach of this problem is based on direction-dependence technique with the heading-angle decision to generate k sub-routes. The reason for the k sub-routes is to evaluate the symmetry of the VPP especially at the beginning of the race. This is not only for the upwind mode and *answer the question of why start to port is better (or not) than start to starboard*, furthermore to evaluate the straight-line trajectory over the rest of the wind modes.

Consequently, k is assigned to have *three values*, 1 for the straight-line trajectory, 2 is for the port(left) and 3 is for the starboard (right) start direction. As a note, on the upwind condition, the straight-line is not evaluated, nor optimized. Further details about k and n are given oncoming sections of this chapter.

Then the minimal time path for the Olympic Laser races is based on equation 4.1 and defined as follow. The objective function is established by equation 4.34 and it is composed of two terms. The first term accounts for the accumulated minimal time up to the previous i *leg* and it stores the k,n state-space variables of the previous legs, as established on equation 4.37. The second term is the minimal time for the current i *leg* from the k sub-route over the n stages.

The time on the second term is determined by the velocity $v_{k,n}$ which depends on its heading-direction, and on the length's trajectory ($dl_{k,n}$). This means that $v_{k,n}$ is conditioned by equations 4.35 and 4.36, and estimated with equation 4.38.

$$\min T = L^{i-1}_{k,n}(\Psi, t_f) + min\Big[\int_{t_0}^{t_f} \frac{dl^i_{k,n}}{v^i_{k,n}}\Big] \quad , k \in \{1,2,3\} \tag{4.34}$$

$$\text{subject to:} \quad \dot{x} = u(\Psi)cos(\Psi) + u_{tw} + u_{tc} \tag{4.35}$$

$$\dot{y} = u(\Psi)sin(\Psi) + v_{tw} + v_{tc} \tag{4.36}$$

where:

$$L^*_{k,n}(\Psi, t_f) = [x_{k,n}, y_{k,n}, t_f]^* \tag{4.37}$$

$$v^*_{k,n} = \sqrt{\dot{x}^2 + \dot{y}^2} \tag{4.38}$$

The boundary conditions showed on equations 4.39, 4.40, 4.41 and 4.42 are determined by the leg, particularly by the location of the start buoy and end buoy. These buoys are located over the sailing area and they determine the direction of the boat to follow along with the direction of the wind, to arrive at the next buoy. The location of each buoy is provided in Cartesian coordinates, as explained in 3.4, the latitude-longitude coordinates can be converted to [X,Y] using the $MATLAB^{®}$ function *deg2utm(Lat,Lon)*. Equation 4.43 defines the initial time for the first leg (*i=1*) which is zero. The initial time for the next leg is given by the final time of the previous leg, as shown in equation 4.44.

$$x_k(0)^i = x_1 \tag{4.39}$$

$$x_k(t_f)^i = x_2 \tag{4.40}$$

$$y_k(0)^i = y_1 \tag{4.41}$$

$$y_k(t_f)^i = y_2 \tag{4.42}$$

$$t_0^1 = 0 \tag{4.43}$$

$$t_0^i = t_f^{i-1} \tag{4.44}$$

To connect all the legs, and predict the optimal time path for the whole race, the next equations 4.45 and 4.46, give continuity to the path. This continuity helps the algorithm to control the tacking maneuver and estimate the tack angle for each leg change.

$$x^{i+1}(t_0^{i+1}) = x^i(t_f^i) \tag{4.45}$$

$$y^{i+1}(t_0^{i+1}) = y^i(t_f^i) \tag{4.46}$$

The tack maneuver refers to the change in direction due to a change of a leg, in other words, it is a transition maneuver. This transition maneuver or the turn/tack angle is the angle between the end section (stage) of one leg with the start section (stage) of the next leg and it is constrained by two equations showed as following and in figure 4.10.
for the tack to port:

$$40° < \Psi^i_{n_{max}-1}(t_f^i) - \Psi^{(i+1)}_{n+1}(t_0^i) < 130° \tag{4.47}$$

and for the tack to starboard:

$$-130° < \Psi^i_{n_{max}-1}(t_f^i) - \Psi^{(i+1)}_{n+1}(t_0^i) < -40° \tag{4.48}$$

The final time on each leg is the minimal time and the result of the optimization over that *i leg* and *k* start direction. However, it is known that on the upwind condition, the Laser follows a zig-zag pattern to move against the wind. Moreover, each of these

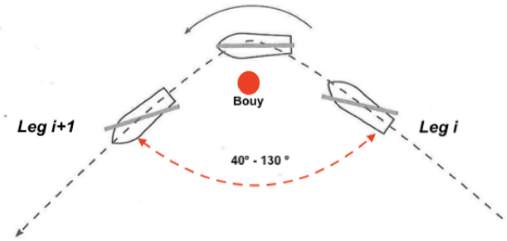


Figure 4.10: Tack angle range between legs.

changes in direction takes time and speed and this has to quantify. For example, [45] mentioned a speed loss of 2 kn due to change in direction, other authors refer to these changes as a delay in time of about 4 to 10 seconds before they are reflected in the trajectory [22], [15].
In this algorithm, the number of changes in the trajectory is quantified as a time loss added to the final time on each leg by equation 4.50. The time loss to add of the equation 4.51 depends on the total number of changes in direction during the *i leg*, set on equation 4.52, multiply by a constant defined as a $t_{tack-loss}$ and its value is defined in further sections on this chapter. In addition, the equation 4.49 controls these shifts in the direction so changes larger than 180 °does not happen.

$$0^\circ <= \Psi^i_{k,n+1}(t^i_f) - \Psi^i_{k,n}(t^i_0) < 180^\circ \tag{4.49}$$

$$e \tag{4.50}$$

$$\Delta T^i_{loss} = t_{tack-loss} \sum_{n=1}^{n} \Delta\Psi^*_k(i) \tag{4.51}$$

$$\Delta\Psi^*_k(i) = \begin{cases} 0, & \text{if: } \Psi^i_{k,n+1}(t^i_f) = \Psi^i_{k,n}(t^i_0) \\ \\ 1 \end{cases} \tag{4.52}$$

$$t^{*i}_f = \sum_{i=1,k}^{i} \left(t^{i,k}_f - t^{i,k}_0\right) \tag{4.53}$$

The state-space limits for the variables are defined in equation 4.54, 4.55 and 4.56. The first two limits the position of the sailboat while the last limits the heading angle (Ψ) to evade the *no-go-zone*.

$$x^{i,1}_{min} < x(t)^{i,1}_n < x^{i,1}_{max} \tag{4.54}$$

$$y^{i,1}_{min} < y(t)^{i,1}_n < y^{i,1}_{max} \tag{4.55}$$

$$\Psi_{min} < \Psi(t) < \Psi_{max} \tag{4.56}$$

$$\text{where:} \quad 40^\circ < \Psi(t) < 320^\circ$$

As mentioned before on each leg the state-space constraints are different, to determine these limits the coordinates of the midpoint of the straight-line trajectory (k=*1*) between buoys are required. The tolerance factor x_{SAtot} and y_{SAtot} determine the minimum and maximum values of the coordinates and defined the limits on each leg. The next equation shows how these estimations are done. In the case of the minimum value or lower bound, it requires the minimum coordinate for X and Y coordinates from the *3 sub-routes* while for the upper limit, it is the maximum of them, in both cases, it is also required the x^i_{mean} *mean values from the 3 sub-routes (k).*

The value of this factor should be chosen carefully, one of the reason is that a bigger area not always has more nodes, and the algorithm will take more time to estimate the optimal solution. Moreover, it is possible that a local minimum will be found instead of the global solution, because of this it was suggested to run the algorithm several times guessing some of the initial conditions. The next iterations serve to tune these conditions until they remain the same [11].

$$x^i_{min} = x^i_{mean} - x_{SAtol} \cdot \left| x^i_{mean} - min\left(x(t^i_0)^i, x(t^i_f)^i\right) \right| \tag{4.57}$$

$$x^i_{max} = x^i_{mean} + x_{SAtol} \cdot \left| x^i_{mean} - max\left(x(t^i_0)^i, x(t^i_f)^i\right) \right| \tag{4.58}$$

$$y^i_{min} = y^i_{mean} - y_{SAtol} \cdot \left| y^i_{mean} - min\left(y(t^i_0)^i, y(t^i_f)^i\right) \right| \tag{4.59}$$

$$y^i_{max} = y^i_{mean} + y_{SAtol} \cdot \left| y^i_{mean} - max\left(y(t^i_0)^i, y(t^i_f)^i\right) \right| \tag{4.60}$$

The optimization problem and its constraints for the minimal time path for the Laser Class have been defined. These constraints not only helps the algorithm to get the optimal solution but also to connect all the legs. The connection between legs is made by means of the tack angle, a simplify approach followed for the purposes of this research [47], [48].

The state-space constraints are coupled not only with the wind forecast model but with the course of the competition. Even when some space boundaries varies according to the leg to course, it is the orientation of leg relative to the wind angle the parameter that tunes the boundaries of this area. For this reason, a further section will explain how the integration of the course is made, more precisely the locations of the buoys into the minimal time path trajectory.

4.3.4. DEFINITION OF THE SUB-ROUTES (*k*=*2,3*) AT THE PORT AND STARBOARD DIRECTION

The optimization algorithm initializes using one of the 3 alternative sub-routes and they are identified by the *k* index. The reason for them is not only because of the heading-angle approach but also because of the *fmincon* function from *MATLAB*® request initial values of the variables used on the cost function to find the optimal solution. Because the problem is non-linear, this means that starting at different points could get different results and these different start conditions facilitate the algorithm to converge to the optimal solution [35], [37].

Starting at different points allows the conversion of the solution, however, in this case, it also aim to the strategy of the race for coaches and athletes. In addition, it aims to limit the area within the optimal path is found and reduces the computational effort also. Because the VPP has a non-convex shape a path attainable region can be defined, the implication of a non-convex VPP implies that the optimal path may not be unique [49],[36]. For example, on the upwind condition, the maximum velocity is found after the 40°, and there are two possible trajectories to follow and reach the next buoy using also the VMG criterion.

First, at 45°the sinus and cosine functions have the same value and it is out of the *no-go-zone*. This means that it is possible to sail at 45°respect to the wind direction until the midpoint and then tack at -45°until the target buoy is reached. By doing these tacking maneuvers on both directions port and starboard directions, the sailing area for each leg can be set up. This area can also be stretched using the x_{SAtol} and y_{SAtol}, this is done to give additional space for possible wind shifts, [40].

The second alternative is to tack at 45°until the target buoy can be reached by making a turn of 90°respect to the wind. This alternative was not followed, first because even when the velocity at 90 °is the fastest the distance to sail is the larger. Moreover, under a constant wind speed and direction, the ratio between the distance to sail versus the ration between velocities is larger. Therefore, there is no advantage to sail at a higher velocity because the distance to cover does no compensate the increment on the velocity and this only increase the time to reach the target buoy.

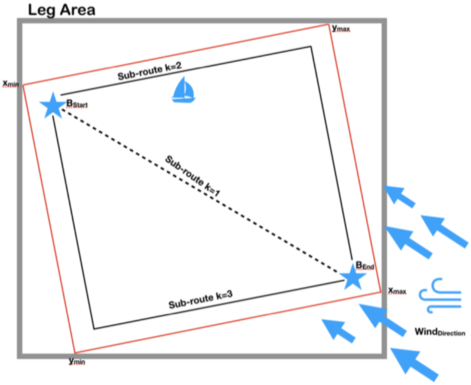


Figure 4.11: Example of a Sail Area for a Leg using the sub-routes

Following the first alternative 2 sub-routes where developed one for the port direction which is assigned by *k=2* and the other to starboard assigned by *k=3*. For the rest of the wind conditions, the same approach is followed to define these two sub-routes and therefore the bounds for the sailing area on each leg.

The n stages of each sub-route are located within each sub-route, so n are points defined by *[X,Y]* coordinates and its location determines the path to follow for a particular leg. In addition to this n stages, a number of intervals between them were defined to describe properly the $dl^i_{k,n}$ of the objective function (equation 4.34). This number of intervals and the n stages are designated at the beginning of the algorithm and they are part of the initialization parameters a large number for the interval value is recommended to have a smooth path, so the interval is designed by equation 4.61. For example, *9 stages* with *100 intervals* it generates *1010* length elements per sub-route.

$$\text{Interval: } m = 100 \tag{4.61}$$

$$elements_{dl} = (n+1)\cdot(m+1) \tag{4.62}$$

These sub-routes serves to define the limits on each leg, not by the means of coordinates of each point but by the maximum and minimum value of those coordinates. The shape of the area for each leg is a rectangle where the coordinates of the opposite corners are defined as equations 4.57, 4.58, 4.59 and 4.60 an example of this area is showed in figure 4.11. In later sections, it will be clear that this sub-routes not only facilitates the algorithm in terms of the space constraints but also in finding the optimal solution for each leg.

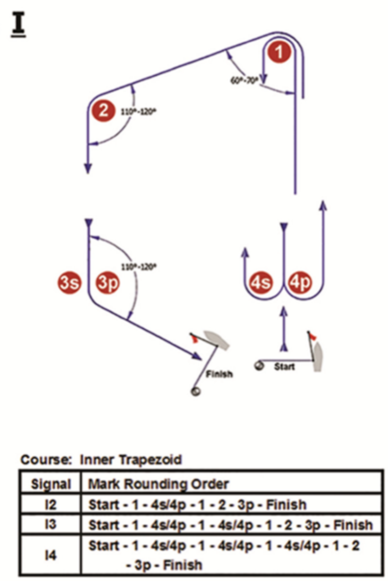
4.3.5. Course Integration into the Time Optimization Path Algorithm

The setup of the course is defined by the organizers of the event and this is done with the diagram and the location of the buoys using *latitude-longitude coordinates*. The importance of them is because the space constraints explained before are linked with the legs of the course which in consequence are related to the buoy's location. The location of the buoys define the distance to sail, the angle direction of the leg respect to the wind direction and as a result, the wind mode to sail.

An example of the information about the course is showed in figure 4.12 and it shows a *trapezoid course*, where the start and end of the competitions are defined by a line. In this course, there are 6 buoys, 2 line indicator and 2 boats. The letter next to the number of the buoy indicates it locations respect to the boat, so *s* is for *starboard* and *p* for *port*. The order of the marks, indicated in the table below of the diagram, describes how the legs are designated and the code or signal for that sequence. The coordinates of the buoys are provided previously to the race since they depend on the wind direction and other factors.

For the purposes of this research, the marks where a line is defined or where 2 buoys are designated with the same number, a midpoint is going to be calculated to represent this condition. However, for the start and the end line, the location of both ends is considered and the midpoint is also going to be calculated. The reason of these three points, on the start and on the end line is to account the variation of the wind direction and review how sensible is the time path on the first leg and on the last leg.

The location of the buoys most of the time is provided in *latitude-longitude coordinates* and the units are degrees-minutes-seconds before to convert them to *[X,Y] coordinates* they need to be converted to degrees. This conversion is made using the *dms2degrees* from *MATLAB*®. Once this conversion is done and using the order of the buoys the algorithm will estimate the distance between them with the *MATLAB*® function *pdist*(b_1, b_2,*'euclidean')*. The angle between them is based on the normal plane, since the wind angle respect to the *North* is determined in degrees by equation 4.63, where i is the index related to the leg. However, even when this expression only refers to the leg, it can be used to estimate the heading-angle(Ψ) and therefore $dl^i_{k,n}$.



Course: Inner Trapezoid

Signal	Mark Rounding Order
I2	Start - 1 - 4s/4p - 1 - 2 - 3p - Finish
I3	Start - 1 - 4s/4p - 1 - 4s/4p - 1 - 2 - 3p - Finish
I4	Start - 1 - 4s/4p - 1 - 4s/4p - 1 - 4s/4p - 1 - 2 - 3p - Finish

Figure 4.12: The I Trapezoid Course diagram provided for the Sailing competitions on Rio 2016 Olympics [3].

$$\theta_i = atan2d\left[\frac{x_2 - x_1}{y_2 - y_1}\right]^i \tag{4.63}$$

The wind mode of the leg is given by the difference between θ_i and the wind angle, β_{tw}, depending on the difference between those angles the wind mode and its angle are defined by equation 4.64. The wind mode angle of the leg (Ω_{i,t_0}) is defined at the beginning of the competition (t_0), in order to start the calculations of the sub-routes. In this case, the β_{tw} only depends on time in the next section is going to be explained how this value is determined since in section 4.3.6 it was explained that it depends on 4 dimensions.

$$\Omega_{i,t_0} = |\theta_i - \text{TWD}_t|, - \begin{cases} \text{Upwind(Upw)} & \begin{cases} 0° & \leq |\theta_i - \text{TWD}_t| \leq 45° \\ 315° & \leq |\theta_i - \text{TWD}_t| \leq 360° \end{cases} \\ \\ \text{Direct(Rch)} & \begin{cases} 45° & < |\theta_i - \text{TWD}_t| \leq 135° \\ 225° & \leq |\theta_i - \text{TWD}_t| < 315° \end{cases} \\ \\ \text{Downwind(Dwn)} & \begin{cases} 135° & < |\theta_i - \text{TWD}_t| < 225° \end{cases} \end{cases} \tag{4.64}$$

The buoys and its orientations respect to the wind define the legs and the limits of the area for each leg using the sub-routes as a reference. The Angle of the Leg respect to the wind (Ω_{i,t_0}) determines the maximum velocity that the sailboat can achieve, once the intensity and location of the boat are known. Now that the parameters for the course on each leg are defined, the next section will be explained how the wind model is going to be integrated into the algorithm.

4.3.6. COUPLING THE WIND MODEL WITH THE SAIL COURSE

Until now, the optimization algorithm has been described along with the state-space constraints only in terms of spatial coordinates. In the other hand, section 3.2 has described the WRF wind model in general, while section 3.3 provides all the details for the model used on this research. To move the sailboat not only it needs a target but also to know the wind velocity V_{tw} at a specific time and location.

This section describes how the wind model of section 3.3 is integrated into this algorithm, so equation 4.38 can be determined. The reasons for this integration and adaptation, first, is because the area covered by the WRF wind model of section 3.3 is much larger than the area of the course. Second, the time step (Δt) is about 10 minutes, meaning that the wind characteristics remain constant along this Δt. However, it is most probably that the space-time dimensions for the estimation of the V_{tw} do not correspond exactly with the space-time coordinates of the grid from the WRF model despite this its value has to be estimated.

Moreover, the time at which the competition starts has to be included during the initialization of the algorithm in addition to the duration of the race and some delays, all these factors are considered for the definition of the *time window*.

Previous to the race date, the location of the sailing course and the estimated time to start it are known. If the conditions are not meet the race could be a delay until this they are meet or in the worst case scenario, the race could be canceled. Because of this, using the start time and the duration of the race (estimated to be about *one hour*). The *time window* is estimated by equations 4.66 and 4.67 adding a bonus time or tolerance up to *three hours* for the upper limit and subtracting *two hours* for the lower limit.

$$\text{Time window} = \left[\text{Time window}_0, \text{Time window}_f\right] \tag{4.65}$$

$$\text{Time window}_0 = \text{Time start}_{floor} - \text{Lower Tolerance} \tag{4.66}$$

$$\text{Time window}_f = \text{Time start}_{ceil} + \text{Upper Tolerance} + \text{Duration}_{Race} \tag{4.67}$$

The tolerances are not the same for both limits because due to the weather conditions most of the time the competition could delay rather than changed for an earlier time. In cases where the time is not defined in hours only, the start time is rounded to a *floor* value, while for the end time it is rounded to a *ceiling* value. The round or floor value, in this case, is set for an integer hour value, but it can be changed for a half-hour or any other value. For example, if the start time is 13:25 hrs, the time window is:

$$\begin{aligned}\text{Time window}_0 &= 13:25\,\text{hr}_{floor} - 2\,\text{hr}\\ &= 13:00\,\text{hr} - 2\,\text{hr}\end{aligned} \tag{4.68}$$

$$\begin{aligned}\text{Time window}_f &= 13:25\,\text{hr}_{ceil} + 3\,\text{hr} + 1\,\text{hr}\\ &= 14:00\,\text{hr} + 3\,\text{hr} + 1\,\text{hr}\end{aligned} \tag{4.69}$$

$$\text{Time window} = \left[11:00, 18:00\right] \tag{4.70}$$

The definition of the *time window* helps to reduce the length of the *time dimension* from the WRF wind model. For example, in section 3.3 it was mentioned the size of their dimensions, which are (x,y,z,t) 198 × 301 × 50 × 145, since Δt = 10 minutes. Using the previous example of the *time window*, this means that instead of using the 145 time-datasets only *43* time-datasets are used on this algorithm. This reduces the time of processing for this model since only approx. 30% of them are used to determine the solution of the minimal time path.

Previous to the competitions, with a couple of months in advance, the location and diameter of the course area are communicated to the participants. Using this information, without any details about the location of the buoys, the area from the wind model for this algorithm is defined as a circle inscribed in a square. The coordinates of the opposite corners of this square are modified as equations 4.71,4.72,4.73 and 4.74 indicate. The tolerance of the wind area is determined by the radius of the sailing course multiplied by n_{sc} times the grid space (Δx) of the WRF wind model.

$$X_{W,min} = X_{CTR,course} - \frac{\text{fi}_{course}}{2} \cdot n_{sc}\Delta x \tag{4.71}$$

$$X_{W,max} = X_{CTR,course} + \frac{\text{fi}_{course}}{2} \cdot n_{sc}\Delta x \tag{4.72}$$

$$Y_{W,min} = Y_{CTR,course} - \frac{\text{fi}_{course}}{2} \cdot n_{sc}\Delta x \tag{4.73}$$

$$Y_{W,max} = Y_{CTR,course} + \frac{\text{fi}_{course}}{2} \cdot n_{sc}\Delta x \tag{4.74}$$

The value of n_{sc} depends on the size of the grid, Δx, in respect to the *radius* of the course as indicated in equation 4.75. Since the [50] establishes that the trapezoid course must be contained in this area, so n_{sc} is the next integer value from the ratio between them. If the coordinates of the buoys are known, the minimum and maximum value of all of them are used instead of the coordinates of the center of the sail area to define the corners of the wind area.

$$n_{sc} = \lceil \frac{\text{fi}_{course}}{2 \cdot \Delta x} \tag{4.75}$$

Using the previous equation the wind area is defined for this algorithm and it is larger than the area of the course, figure 4.13 sketch this concept. It shows that if the center of the sail area is sufficiently far for any grid data-point, for example, within a sail of area with a fi $= 3\Delta x$ it only contains *four* data-points while within the wind area defined with the previous equations it contains 64 data-points.

It is important that regardless of the location or time, the algorithm should be capable to estimate the wind's velocity. Particularly when the coordinates are not coincident with those of the grid from the wind model. Thus to estimate the velocity at any point over the space and time an interpolation method is required. The data-points around the point of interest must be enough to estimate this value, moreover the area defined for the wind.

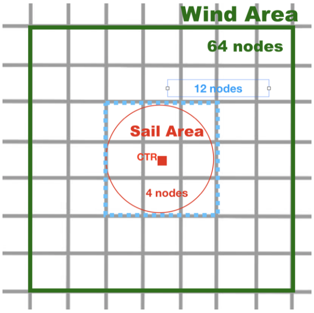


Figure 4.13: Wind Area Concept using a wind model representation. The inscribed sail area shows the limits of the square and its corners are used for the definition of the wind area.

The interpolation method used in this algorithm is the *griddata* function from $MATLAB^{®}$. This function-method estimate the components of the wind's velocity using the coordinates from the grid arrangement of the WRF wind model. Figure 4.14 shows a graphical representation of how this function and the wind model interact. It shows how two datasets for the velocity with coordinates *[X,Y,t]* are used to determine the velocity V^*_{tw} at t^* which is intermediate from t_0 and t_1. Inside the function to obtain the requested value, a method has to be defined in cases where a linear interpolation doesn't apply. For the velocities, in this research, the *nearest* option for the method is used inside the *griddata* function.

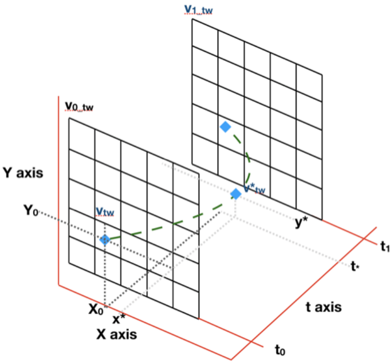


Figure 4.14: Griddata Method, a graphical representation of two datasets. The function interpolates the coordinates and values from the grid to obtain an intermediate value

The coordinates used along this algorithm are *[X,Y,t]*, this means that any data using *[Lat, Lon]* coordinates have to be converted to *Cartesian coordinates* (*[X,Y]*), such as the buoys coordinates. In the case of the time dimension (t), the units of the time step are defined in minutes, and this has to be converted to seconds, moreover, the number of digits after the point is only one.

The minimal time path algorithm not only requires the characteristics of the boat and athlete to solve the problem, but also constraints. These constraints are related to the state-space variables and they embody the environment within the competition takes place. In addition, to the type of maneuvers commonly used by athletes and sailor to sail from one point to another. But to represent properly the environment the area to sail has to be coupled to the wind area so the minimal time path can get a solution.

Now that the algorithm has been defined and most of their parameter also, the next section will validate them. At the same, it will review that the parameters defined such as the number of stages is acceptable. All these to verify that a path developed using this algorithm represent the typical path of the laser class developed during races.

4.4. Algorithm Validation: Results and Considerations

The purpose of this section is to verify and validate the functionality of this algorithm to estimate the time of the predicted paths, thus to determine the minimal time path. These paths have to meet the constraints previously established and at the same time evaluate the values for the parameters assigned. Particularly the parameter for the number of stages per leg (n) when the wind mode to sail is *upwind*. In addition, to the constant related to the tack loss time ($t_{tack-loss}$) added to each of the shifts on directions of the laser class. By the end of the section, the considerations, and adjustments on the model required are mentioned in order to solve the optimization problem.

The validation is made on an upwind mode because during this mode the laser class is prone to follow a zig-zag pattern. As a result of this, it was defined that the wind speed (V_{tw}) and direction (β_{tw}) to be constant over time and space. In the other hand, the number of stages to evaluate begins at one (*(n=1)*), so one shift in direction is made to reach the target.

The setup of the parameters and conditions represent the most simplified conditions to evaluate the algorithm but it can reveal easily if the algorithm works properly or not. The first aspect to review is that the algorithm identifies the *no-go-zone*, for this the distance between marks is *0.96 nm* equivalent to 1730 meters and $\Delta\Psi$ is 5°. Using these parameters, figure 4.15 shows that the algorithm develops more than. 346,000 nodes over the whole area defined inside a rectangle of approx. $8000 \times 4000 m$; however, many of these points are located above the target's location. The marks on green, only at one side of the start point are the nodes that meet the criteria about the angle and the *no-go-zone*. The other side was not plotted since the VPP is assumed symmetrical.

The values for the parameters and conditions to validate the algorithm are shown below. In this case, only half of the angles from the VPP($[0,\pi]$) was used since it is assumed that it is symmetrical. Moreover, it is assumed constant wind conditions so the paths on both sides from the vertical line on the start mark are symmetrical. These parameters are:

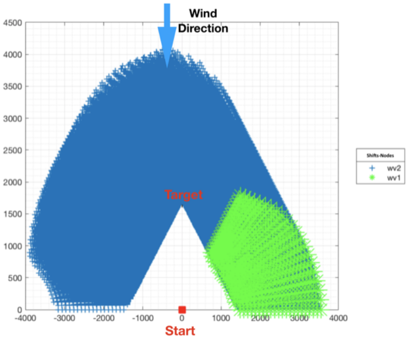
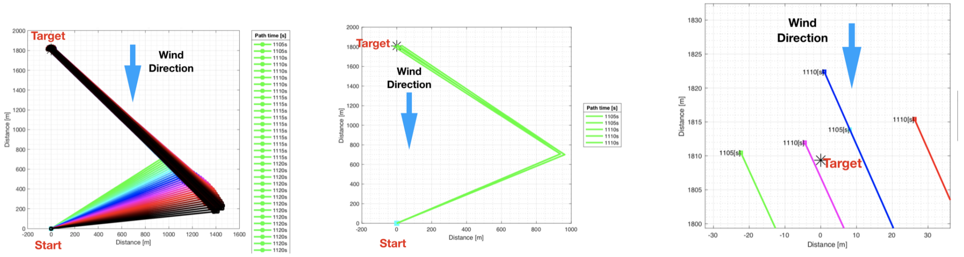


Figure 4.15: Nodes generated between 2 marks with one stage *(n=1)* and $\Delta\Psi = 5°$

- V_{tw}= 8.5 kn equivalent to 4.37 m/s.
- β_{tw}= 0°from *North*.
- The distance between the start line and the next mark is *0.977 nm* equivalent to 1809 meters.
- The number of points stages is two, *(n=2)*.
- The time step (Δt) is 5 seconds.
- The heading step angle $(\Delta\Psi)$ is 1°.
- The $t_{tack-loss}$ = 10 seconds.
- The tolerance factor for the leg area are for x_{SAtol} = 2.5 and for y_{SAtol} = 2.

Under these conditions, more nodes were generated compared with figure 4.15, and in consequence more paths with even the same time. This is shown in figure 4.16a where paths inside the same time range were plotted with the same color. The figure also shows that the fastest paths do not shift after 1200 meter from the start point. In fact, figure 4.16b shows that the paths with the top 5 times shifts before a horizontal distance of 1000 meters, which is only 10.5% more than the midpoint distance between the marks. Despite the number of paths and its times the target mark hasn't reach perfectly this is shown in figure 4.16c, where the second best time is the closets path within a radius of 6 meters from the target mark.

When the number of points-stages increase to two (*n=2*), not only the number of nodes and paths increase but the top 5 times also increase by 5 seconds. In this case, the first shift in direction occurs at a horizontal distance from the start mark of 1100 meters and following the same heading-angle, figure 4.17a. The second shift on direction occurs within a radius of 46 meters from the target mark as shown in figure 4.17c. But despite all these alternatives, the end of each of the paths does not reach perfectly the target mark. Figure 4.17b shows that all the paths with the same time end within a radius of 3 meters from the target mark.

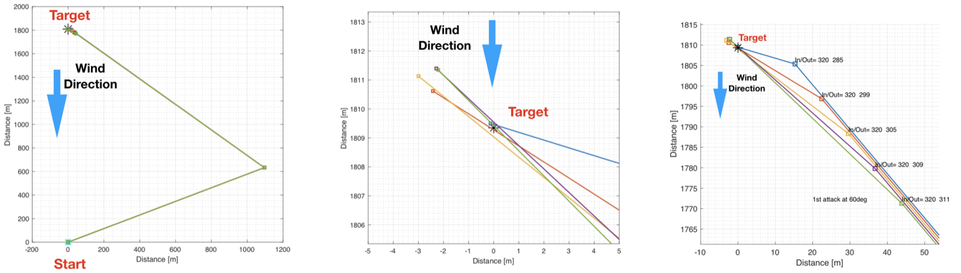


(a) Paths generated with *n=1* and $\Delta\Psi = 1°$.

(b) Top 5 time paths and closets to the marked. These paths have the same time.

(c) Minimal time paths close-in end at the target mark.

Figure 4.16: Paths generated with one shift point in the direction (*n=1*) and $\Delta\Psi = 1°$.



(a) Top 5 times using 2 attacks with Δt =5s and $\Delta\Psi = 1°$. The time for all the paths is 1100s

(b) Close-up of the top 5 times at the target mark. The ends of paths are within a radius of 3m from the target mark.

(c) Details about the shifts in the direction after the first heading-direction was taken.

Figure 4.17: Top 5 time paths generated with two shift points in the direction (*n=2*) and $\Delta\Psi = 1°$, the time of all them is 1100 seconds.

From the previous results, it was clear that another constraint or another adjustment to the algorithm are required to eliminate this kind of solutions. One of these changes refers to the definition of *n*, which is the number of points inside each leg in order to define the number of internal stages within it. To represent this *n* is a point with coordinates *[X,Y]* inside a vector where its first set of values represent the coordinates of the start buoy and the last element is the coordinates of the end buoy. This means that *n* is the number of points inside a vector and regardless the coordinates of these points they are linked to the legs using the buoys. It can be said that *n* defines the size of a vector for the stage-state variables as in equation 4.76, this vector define the number tacks or changes in directions that path could have and in this way it could develop a zig-zag pattern. For example figure 4.17 is a path where n=2, this means that between the start and end point the path could have two changes in direction.

$$\text{stage-state vector} = \begin{bmatrix} x_{bouy,start} & y_{bouy,start} \\ x_1 & y_1 \\ \vdots & \vdots \\ x_n & y_n \\ x_{bouy,end} & y_{bouy,end} \end{bmatrix} \tag{4.76}$$

The value of n stage points in this research is assigned to be *7*, this means that inside each leg there are 8 stages where a shift in direction could occur. Besides, at least 70 seconds could be added to the time's path if these shifts occur along each leg. The locations of these n points are determined by the space constraints assigned to each leg and by the angle constraints, in other words by equations 4.57 to 4.60.

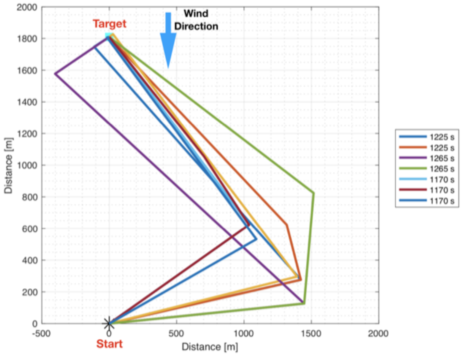


Figure 4.18: Paths generated by a wind field uniform and constant coming a 0º North using the stage-state vector n=*2* and $\Delta\Psi = 1°$. The optimal time paths are two with 1225 seconds and with different trajectories.

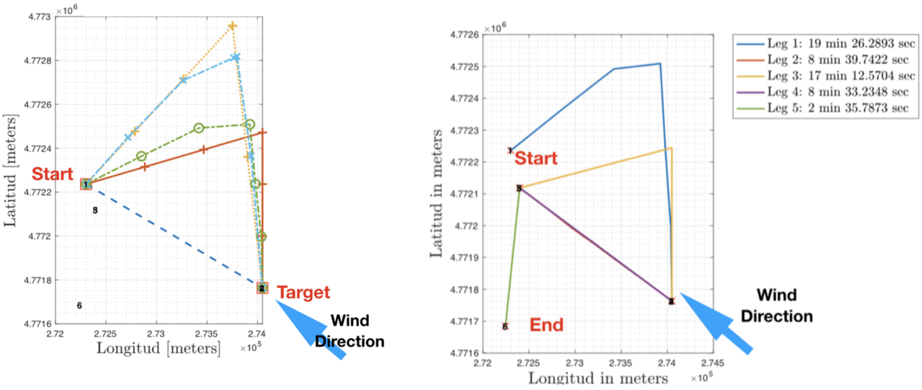
Applying these changes figure 4.18 shows some of the paths developed with them. It also shows that the fastest paths make the first shift before 1050 meters from the vertical axis of the start mark. Furthermore, this distance is 1.15 times the midpoint distance between marks and this ratio is the same when n=*2*, and equations equations 4.57 to 4.60, use a factor (x_{SAtol} and y_{SAtol}) to scale up the area for each leg. Using these trials equation 4.77 indicates the value of these tolerance factors, for now, it is assumed that the value for each factor is the same for both coordinates.

$$x_{SAtol} = y_{SAtol} = 1.15 \tag{4.77}$$

Applying all thee changes the algorithm can developed paths that meet the constraints and estimate its time. For the optimization problem, particularly for the *fmincon* function, where the option for the internal algorithm has to set as *'active-set'*. This option allows the algorithm to take large steps far from the target so the number of iterations and evaluations of the algorithm are used effectively to find the optimal solution. It was observed that without this change, the solution takes a larger number of iterations and evaluations of the function to get the optimal solution. Besides, most of the time this solution is not even the optimal compared with the solutions provided with the *'active-set'* method.

After making these changes in the algorithm and adding a trapezoidal route as figure 4.12, with the same V_{tw} as previously defined but with β_{tw} about 140°. The optimal path for the race was found it and it is shown in figure 4.19b, where each leg is indicated by a different color, the total time is about 56 minutes. Because the wind is constant in time and space the fastest trajectory in a downwind and direct wind mode is described as a straight line between the buoys.

The optimal path for the upwind mode on leg 1 was found by developing the 2 sub-routes described on section 4.3.4, these sub-routes represent different start conditions. These two conditions were used by the algorithm to be optimized and the minimal time between them provides the optimal path for the leg. Figure 4.19a shows 4 paths, two of them are the start conditions, the sub-routes *k=2* marked in red and *k=3* marked in yellow. The estimated time for *k=2* is *20.84* minutes while for *k=3* is *21.46* minutes, the optimization from *k=3* is marked in blue and its time is *20.59* minutes. The green trajectory is the optimal path between them with *19.43* minutes and it was used to construct the optimal path for the rest of the race since it was used on the angle constraint with the next leg.



(a) Leg 1, Upwind Mode

(b) Race Optimal Path per leg

Figure 4.19: Trapezoid Course Optimization Time Path Solution

The optimal time path or the minimal time path can be found by finding the minimal times per leg, the initialization of the algorithm requires an initial guess for the variables used. Because the system is non-linear due to the no-convex shape of the VPP the optimal solution is not unique. The algorithm developed considers two initial guess values for the variables (sub-routes *k*), after their optimization their results are compared to find the minimum time path between them. The algorithm uses the heading-direction approach to find the optimal path, and it divides each leg into stages.

These points also control and limits the number of shifts in direction that define the zig-zag pattern specifically in the upwind mode. Furthermore, the points-stages *(n)* inside the leg enable the continuity between legs, this continuity is constrained by the angle between legs and between these points. The locations of these points are limit by the area assigned to each leg since the space area determines the number of nodes or coordinates where these points can be located. Most important, the area must be contained inside the wind model area and it has to be large enough to estimate the wind's velocity (V_{tw}) and direction (β_{tw}) at any time and location.

Now that the algorithm was validated to developed paths, estimate its time and optimized them to get the minimal time path the next chapter will evaluate 2 conditions related with the time-step. All the details about the conditions and parameters will be provided in the next chapter.

5

Results for the Race 1 at the World Cup Series 2018 at France

Sailing regattas for Olympics classes take place all around the world and one event that features them is the World Cup Series. This competition is a series of regattas organized at various cities around the world during a year and it is the Word Sailing Organization responsible to regulate this event. This means that the races follow most of the policies that govern the Olympic Sailing Events. One series of the World Cup 2018 was hosted by Hyéres, France during April.

SAP Sailing Analytics® recorded all the details of each race, including participants, times, GPS trajectories and wind measurements and it is open access [51]. Because of this, one race during this event is a reference for the estimated results using the algorithm developed on this research and for the WRF wind model developed.

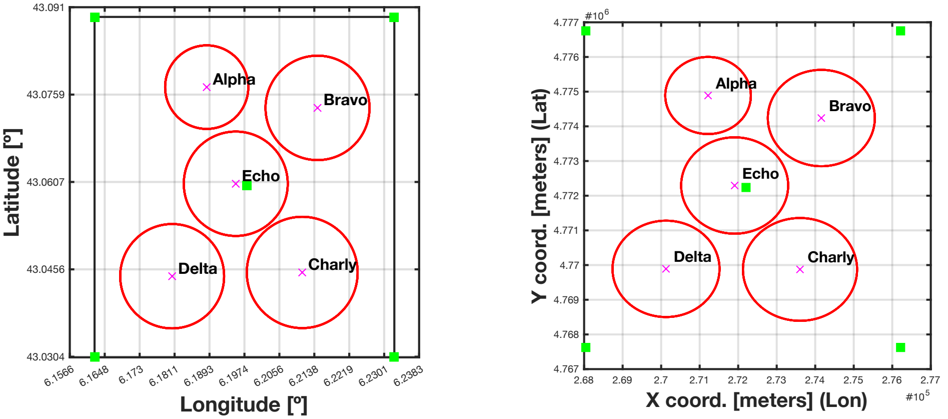
The aim of this chapter is to compare the optimization results, time and trajectory, for a laser race from 4 configurations for the wind model. The first configuration uses only the velocity and direction of the wind. The second model assumes a step time of 1 hour, this means that all the wind properties remain constant for 1 hour. The third assumes a step time of 10 minutes (*1/6 hr*) and for the last configuration, the algorithm uses the wind measures from [51]. In each configuration, the legs under the upwind mode compare the results from the optimization when the heading-direction is the port and when it is at starboard.

This chapter is organized into three sections, the first section describes the location and area of the race with the parameters mentioned during the previous chapter. The next section shows the inclusion of the WRF wind model area with the course area and last part presents the results of the tests described before. The order of these results follows the same order introduced in this chapter.

5.1. Race Integration and Initialization Parameters of the Algorithm

The laser race to simulate took place in the World Cup Series 2018 in Hyères, France during April. The identification label of the race is *R1*, and it is a trapezoid course with 5 legs, 2 of them sailed at upwind-downwind mode and the last under direct wind mode. The leg is the section of the race defined by two buoys. The event features all the Olympic classes organized in 5 areas around the bay as shown in figure 5.1, the area within R1 took place in *Echo* and its diameter is approximately *1.5 NM* equivalent to 2963 meters, with its center at 43º 04.144'N,006º 11.913'E.

The wind area defined uses the parameters from section 4.3.6 and the courses areas to limit the wind model area before knowing the locations of the buoys. Figure 5.1a shows the locations of the areas for the competition in *Lat-Lon coordinates in degrees* In the figure the red lines are the limits of each course and the green squares denote the limits for the WRF wind model used during this simulation, this means that a portion of the WRF is used. The green square at the center shows the center coordinates from all the courses areas. First, the algorithm uses the area inside these limits, and once the buoys coordinates are available, then it defines the area of each leg which limits the location of the stage-points at each leg.



(a) Courses Areas Locations using Lat-Lon (deg)Coordinates.

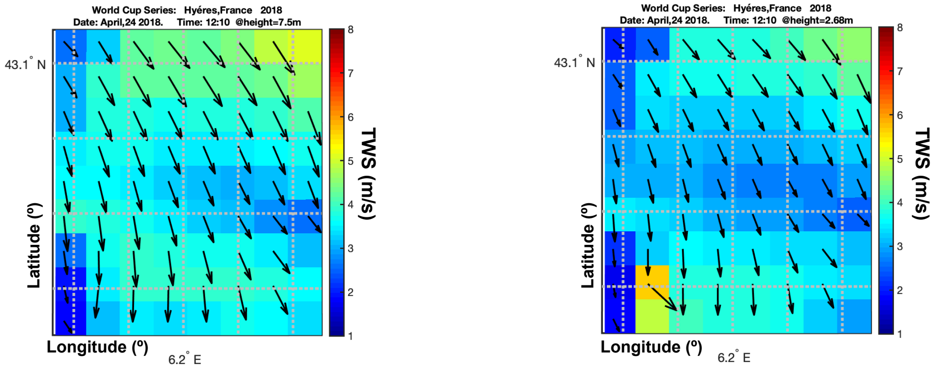
(b) Courses Areas Locations using X-Y Coordinates (m).

Figure 5.1: Courses Areas of the World Cup Series 2018 Hyéres, France

The courses of the competition adjusted the wind model area in the *XY coordinates*, for the height (*Z coordinate*) this optimization algorithm only considers the first two levels. Their corresponding heights, in meters, are *7.5* and *25*. Because, the velocities for the optimal path are at the CE, at a shorter height, and they are getting using equation 2.1 and the two heights mentioned before.

First, the calculation of the CE for the laser's sail uses the *40%* of the height's sail and then adds the height of the sail foot, from the top of the deck [34],[11],[16]. As a result, the CE is at *2.68 meters* from the sea level. Figure 5.2 shows how the height influences the velocities, both figures are over the same area and time. The figure uses *Lat-Lon coordinates*, and it includes all the courses shown in figure 5.1a, the black arrows show the divergence of the wind field and the direction from where it comes.

The velocities at *7.5 meters*, 5.2a have a larger magnitude at the center of the area than those at *2.68 meters* where the magnitude in average is smaller, except for the left-bottom corner of figure 5.2b. The reason for such behavior is that of the power relation (exponent κ of equation 2.1. This equation describes the ratio between heights at power κ, to the ratio between their velocities, therefore, the wind behavior at that area does not appear at 7.5 meters.


World Cup Series: Hyéres,France 2018
Date: April,24 2018. Time: 12:10 @height=7.5m
43.1° N
Latitude (°)
Longitude (°)
6.2° E
TWS (m/s)
1
2
3
4
5
6
7
8
World Cup Series: Hyéres,France 2018
Date: April,24 2018. Time: 12:10 @height=2.68m
43.1° N
Latitude (°)
Longitude (°)
6.2° E
TWS (m/s)
1
2
3
4
5
6
7
8

(a) WRF wind map at 7.5m height

(b) WRF wind map at 2.68m height

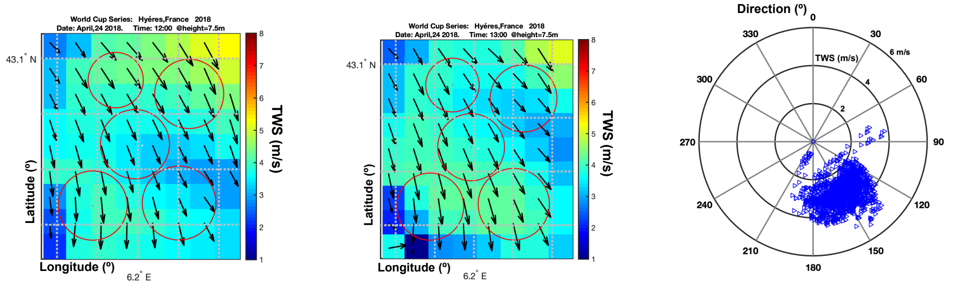
Figure 5.2: Wind map at 2 different heights over the Courses Areas of the World Cup Series 2018 Hyéres, France

Another dimension to consider is the time because the same area could be subject to distinctive wind patterns as time went by. For this, the algorithm requires an estimated time at which the race could start and use equation 4.65 to set the *time window*. The *time window* describes the distribution of the V_{tw} and its β_{tw}. Assuming that the competition started at *12:00 hrs*, the equation 4.65 defines the *time window* as follows.

$$\text{Time window} = [10:00, 16:00] \tag{5.1}$$

At 7.5 meters height the range of velocities and angles estimated for the race, which has a duration approximately of one hour, is less than 6 m/s (11.66 kn) while its angle range is between [90,180] degrees with dominated directions from [120,180] degrees. Figure 5.3c shows both ranges, the circles represent the velocities, the bigger the circle, the larger the velocity.

The distribution of the V_{tw} over the map on the courses areas, figure 5.3a and 5.3b, show that the central area for the race has a velocity range between 3 m/s to 5 m/s, also the model WRF predicts to have higher velocities one hour after the race starts. Thus, the maps show the distribution of the wind at that specific time while figure 5.3c involves all the values and changes from *12:00*hrs to *13:00*hrs. The 7.5 height is a reference because for wind measurements this is the minimum height recommended and the most used.


World Cup Series: Hyéres,France 2018
Date: April,24 2018. Time: 12:00 @height=7.5m
43.1° N
Latitude (°)
Longitude (°)
6.2° E
TWS (m/s)
1 2 3 4 5 6 7 8
World Cup Series: Hyéres,France 2018
Date: April,24 2018. Time: 13:00 @height=7.5m
Direction (°)
0 30 60 90 120 150 180 210 240 270 300 330
TWS (m/s)
2 4 6 m/s

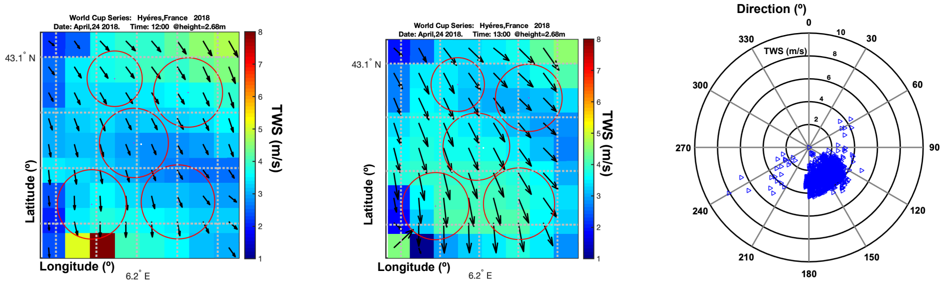
(a) WRF Wind map at 12:00 hrs.

(b) WRF Wind map at 13:00 hrs.

(c) Wind Directions from 12:00 to 13:00 hrs.

Figure 5.3: WRF Wind Area at 7.5m with the courses from the World Cup Series 2018 Hyéres, France

At the CE height, the wind velocities pattern and angle range show similar outcomes than at 7.5 meters contrarily the velocities at this height are smaller. The speeds estimated at the end of the race are higher than initially similarly, figure 5.4b shows the wind field is also higher at the end. The beginning of the race, figure 5.4a, shows 2 areas with higher velocities and the central area of the competition has a wind speed around 3m/s.


World Cup Series: Hyéres,France 2018
Date: April,24 2018. Time: 12:00 @height=2.68m
43.1° N
Latitude (°)
Longitude (°)
6.2° E
TWS (m/s)
1 2 3 4 5 6 7 8
World Cup Series: Hyéres,France 2018
Date: April,24 2018. Time: 13:00 @height=2.68m
Direction (°)
0 30 60 90 120 150 180 210 240 270 300 330
TWS (m/s)
2 4 6 8 10

(a) WRF Wind map at 12:00 hrs at CE height.

(b) WRF Wind map at 13:00 hrs CE height.

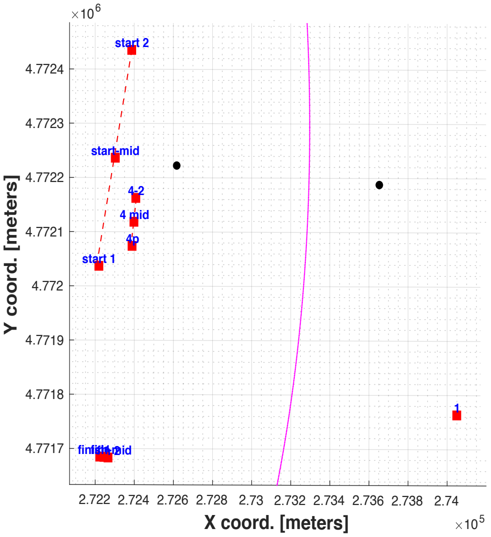
(c) Wind Directions from 12:00 to 13:00 hrs at CE height.

Figure 5.4: WRF Wind Area at CE height within the courses from the World Cup Series 2018 Hyéres, France

In fact, the range of speed velocities expected (V_{tw}) during the race, from 12:00 to 13:00 hr, is close to 4m/s similar to the ones at 7.5 meters. However, figure 5.4c shows that at this height some areas experience larger velocities compared with the velocities at 7.5 meters and with various directions, the angle range is from [60,240] degrees. The concentration of the speed and angles is equivalent to the 7.5-meter height.

After completing the adaptation of the wind area to the course areas from the World Cup Series 2018 in Hyères, France, and define the time window, the simulations only requires the locations of the buoys. The buoy's locations are in *XY coordinates*, instead of *Lat-Lon coordinates* hence the velocity and displacement use the same base units.

The race to simulate is the *R1*, and it has a trapezoid shape with 5 legs. The leg is the section of the race defined by two buoys and they are indicated in the table on figure 5.5. The leg 1 and 3 are in the upwind mode, both finished at buoy 1 but started at other location. The WRF wind area used is larger than the area covered by the buoys as shown in figure 5.5, the red squares are the buoys locations with its identification label in blue. The magenta line represents the limit of the *Echo* course, previously explained, and the 2 black dots are the locations of the grid points where the wind model estimates the wind's velocity ($V_{tw_{i,j}}$).

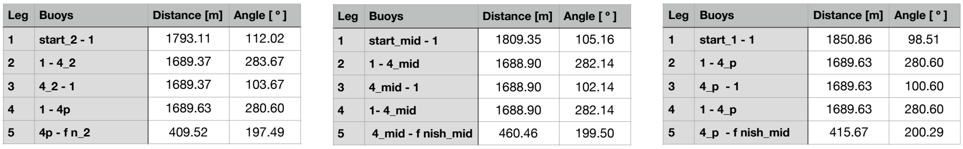


Leg	Buoys
1	Start - 1
2	1 - 4p/4_2
3	4p / 4_2 -1
4	1 - 4p
5	4p - Finish

Figure 5.5: Buoys location in *XY coordinates* for R1 for the Laser Class at the World Cup Series 2018 in Hyéres, France

The figure shows 2 lines a start line and a second line that uses the buoys labeling with the number 4, and a third smaller line for the finish buoys at the bottom of the figure. This algorithm uses also the midpoint of all the lines to estimate the optimal path. The start line is the longest of them, 430 meters while the end line is the shortest with 42 meters, and the other line is about 90 meters. Because of the configuration of the racecourse, and these lines the length of the first leg varies from 1793 to 1850 meters with and its angle goes from 98.5°to 112 °respect to the North. To find the optimal path, the algorithm discretizes the lines using 3 points the north, the middle and the south point. The length of each leg and the angle direction for each point configuration are in figure 5.6.

The race course is smaller compared with the area defined for the wind and to estimate the average wind properties to which the race is subject to, the limits for the race have to include more grid points. A visual representation of the wind characteristics and the racecourse in *XY coordinates* on meters is in figure 5.7. To have enough information about the wind over the racecourse, the number of grid points is at least 12. Using these grid points showed by arrows on figure 5.7a the wind properties estimated, V_{tw} and β_{tw},



Leg	Buoys	Distance [m]	Angle [°]
1	start_2 - 1	1793.11	112.02
2	1 - 4_2	1689.37	283.67
3	4_2 - 1	1689.37	103.67
4	1 - 4p	1689.63	280.60
5	4p - fin_2	409.52	197.49

(a) Using the north points of the race lines.

Leg	Buoys	Distance [m]	Angle [°]
1	start_mid - 1	1809.35	105.16
2	1 - 4_mid	1688.90	282.14
3	4_mid - 1	1688.90	102.14
4	1- 4_mid	1688.90	282.14
5	4_mid - finish_mid	460.46	199.50

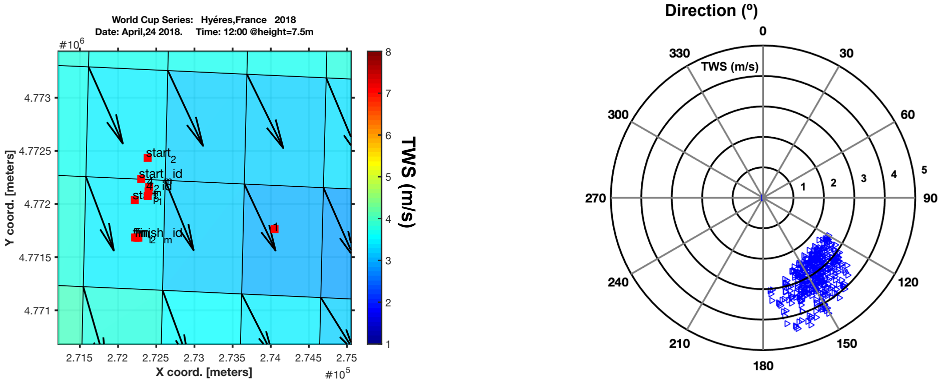
(b) Using the middle points of the race lines.

Leg	Buoys	Distance [m]	Angle [°]
1	start_1 - 1	1850.86	98.51
2	1 - 4_p	1689.63	280.60
3	4_p - 1	1689.63	100.60
4	1 - 4_p	1689.63	280.60
5	4_p - finish_mid	415.67	200.29

(c) Using the south points of the race lines.

Figure 5.6: Length's leg and angles using the respective points of the each race line.

are 4m/s with a wind range angle about [120,180] degrees, figure 5.7b. The figure also shows that a straight-line path for leg 2 using these limits is now subject to 3 diverse grid points. Because of these, for the optimal path prediction the algorithm uses more points.



(a) Close up to the Buoys location for R1.

(b) Wind Directions from close up to the buoys location for R1.

Figure 5.7: Buoys location in meter using *XY coordinates* and the WRF wind model for World Cup Series 2018 Hyéres, France

After defining the racecourse with its legs and its location respect to the wind area, the following section shows the results for the simulations performed. The sailing races for Olympic classes are subject to changes, however, with the information previously provided is possible to limit the area and define the time window. The locations of the buoys tunes this setup and parameters, for example, the wind conditions can change the locations of the racecourse at areas not considered initially by the courses areas, hence, the locations of some buoys are out of the initial course area. Changes on the start time and location of the racecourses happen frequently during the development of these events. Despite these changes, which could occur one day before the competition, they don't have a negative impact on the definitions of the model. Since the last changes or information provided contributes to locate precisely the race and then define the space limits on each leg.

5.2. THE WIND SCENARIOS FOR THE TIME OPTIMIZATION ALGORITHM

This section shows the results of five scenarios where the time step and space dimension have distinct configurations. The first and simplest scenario assumes a uniform and constant wind field, the rest of the scenarios varies the time step and grid configuration. The minimal time path for the race results from the minimal time path for each leg. Therefore, the algorithm analyzes the legs configurations from figure 5.6.
On each leg, it also tests the tack direction at the start, to port or to starboard at various angles. These means that the algorithm on each leg uses at least 12 initial conditions for its optimization. The reason for them is not only to review the convergence of the solution but also to eliminate the local optimal solutions. The legs under upwind are the most challenged since they are the longest and their trajectory can't follow a straight line. For example, under the upwind mode, regardless of the wind model used, the algorithm identify six minimal time paths, two per each initial star location from figure 5.6.

The minimal time paths generated according to its start location are one for the port direction and the other to starboard. Figure 5.8 indicates each of the minimal times paths generated using different colors, the buoys are indicated by red squares and next to it are their names. The figure also include the time for the path. Moreover, the two best paths, the ones with the smallest time are indicated by a thicker line.

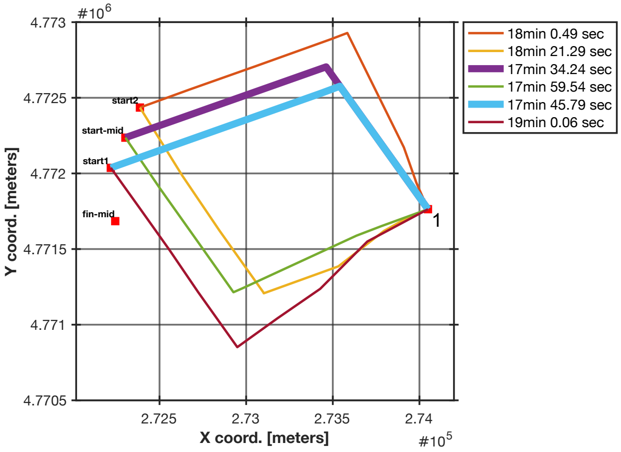
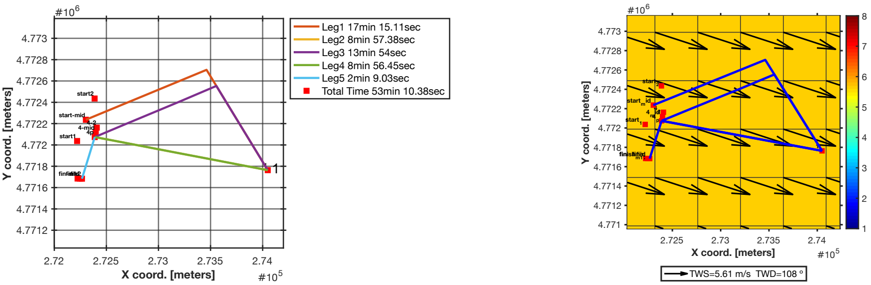


Figure 5.8: Paths and its times generated by the algorithm for the upwind-mode using 3 different start locations. The two best times are indicated by a thicker line.

The rest of the legs follow the same process, at the end the algorithm presents a figure with the optimal time-path by leg, its time and the total time. On this section the figures to show are like figure 5.9 with the optimal time-path for the race where the times by leg are included and figures about the wind field at the beginning and/or at the end of the race, depending on the scenario.

The first scenario is a constant and uniform wind field. The second has a time step of 60 minutes or 3600 seconds, this means that the wind conditions remain constant



(a) Total Time and Time per Leg using a constant and uniform wind model.

(b) Wind Field Conditions for the beginning and end of the race

Figure 5.9: Optimal Time-Path trajectories and times per leg using a Constant Wind Field of 5.61 m/s at 108 °

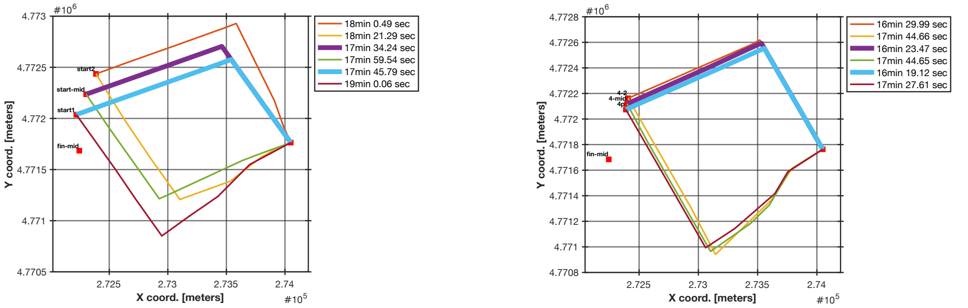
during this time but varies over the space. The last scenario uses the full resolution of the WRF model, the time step is about 10 minutes or 600 seconds and it varies over the space with a grid size of 1km in *XY coordinates*. The results from the scenario using the wind measurements from the race taken at 5 locations with a sampling rate of 20 Hz is at the end of the section since the space distribution of the model does not match with the WRF. The race started at 12:11 hrs on April 24, 2018; this time defines the initial conditions of the wind while the algorithm with the buoys coordinates determines the space constraints for each leg.

5.2.1. CONSTANT AND UNIFORM WIND FIELD

The properties of this scenario are the simplest used in this research. The value of the V_{tw} was 5.61 m/s or 10 kn with and β_{tw} equal to 108 °respect to the North. The reason of this scenario is because most of the forecast for public access provides the velocity and direction valid for one hour within a space resolution larger than the course area. Figure 5.10 shows the two legs on upwind mode, Leg 1 and Leg 3, on these legs the start position uses the line discretization represented by 3 points. Another parameter is the tacking direction at the start, to port or to starboard, these means there are at least 12 initial values of the variables to optimize.

The figure shows the best solution by start location and by direction, this means that there are only 6 paths presented. The two optimal time paths for the leg 1 and 3 , on upwind mode begins at the center and south position of the start line. These paths are indicated on figure with thicker lines.

For this scenario and on the upwind mode, the minimal time path follows one tack direction after the start and it goes to the North but the star point is at the middle point of the start line and alternative start is at the south point. For the rest of the legs, the straight line gives the minimal time path, figure 5.9a shows the minimal time path for the race and the conditions used in this scenario. In this scenario the minimal time path is 53 minutes with 10.38 seconds, the first leg starts at the midpoint while the third leg is at the south point, the angle of tacking to arrive at the next buoy is the same for both legs. The times between the leg 1 and leg 3 differ by almost 4 minutes even when both



(a) Minimal Time Paths for Leg 1 on upwind-mode for each of the star positions when the wind model is constant over time and space.

(b) Minimal Time Paths for Leg 3 on upwind-mode for each of the star positions when the wind model is constant over time and space.

Figure 5.10: Upwind Legs 1 and 3 with Times per Leg and trajectories generated according to its position on the starting line and direction using a Constant Wind Model. The two minimal times by leg are indicated by thicker lines.

follows the same trajectory at the end.

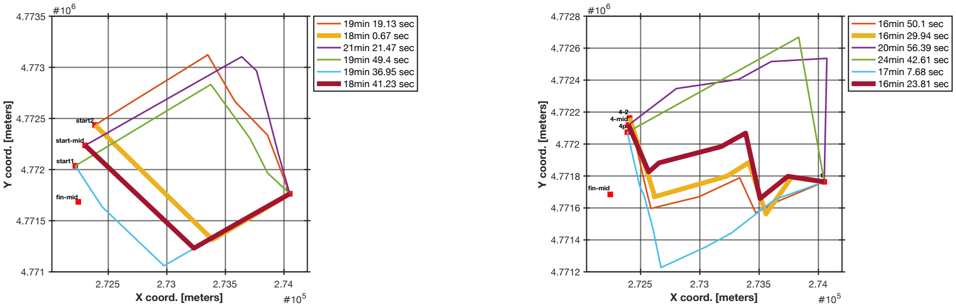
5.2.2. Wind Field with Time Step of 60 Minutes (1 Hour)

In this scenario, the wind field is from the WRF model, the space grid is 1 km and it remains constant for 60 minutes. This scenario reduced the computational effort since it omits the time dependence because the race has a duration of one hour at the most. The legs on the upwind mode, for this case, shows an alternative direction for the minimal time trajectory compared with the previous scenario.

For the leg one, the initial tacking goes to the south and the start point is the midpoint and the one in the north. The leg three in this scenario has more tacking maneuvers and the best path goes to the south also, figure 5.11 shows four significant changes in the sailboat's direction. Its pattern follows a zig-zag pattern coming from the south.

The times for both legs on upwind mode for this scenario are larger than in the previous scenario even when the wind conditions reviewed are at 12hrs and at 13hrs. The minimal time trajectory is 54 minutes and 21.97 seconds and figure 5.12 shows the trajectories for each leg, its time and the total time . It also indicates that the optimal time-path for the upwind leg 1 starts at the North point while the leg 3 starts at the center, point 4-m.

The difference of the wind field between the start time and end, are in figure 5.13 where both intensity and direction in average have changed. The direction of the wind has an angle deviation close to 5°and at the end, the wind velocity is larger than at the start. The south part of the graph at the end of the race has larger speeds than at the North, while at the start of the race the North had higher wind velocities. This shows how much the velocities can vary over one hour. The figure contains the optimal time-path for the race represented by the blue line connecting the red squares which are the buoys of race.



(a) Minimal Time Paths for Leg 1 on upwind-mode for each of the star positions when the wind model is constant for 60 minutes and with a space grid on 1 km.

(b) Minimal Time Paths for Leg 3 on upwind-mode for each of the star positions when the wind model is constant for 60 minutes and with a space grid on 1 km

Figure 5.11: Upwind Legs 1 and 3 with Times per Leg and trajectories generated according to its position on the starting line and direction using a Wind Model Constant constant for 60 minutes over a grid of 1 km per side. The two minimal times are indicated by thicker lines.

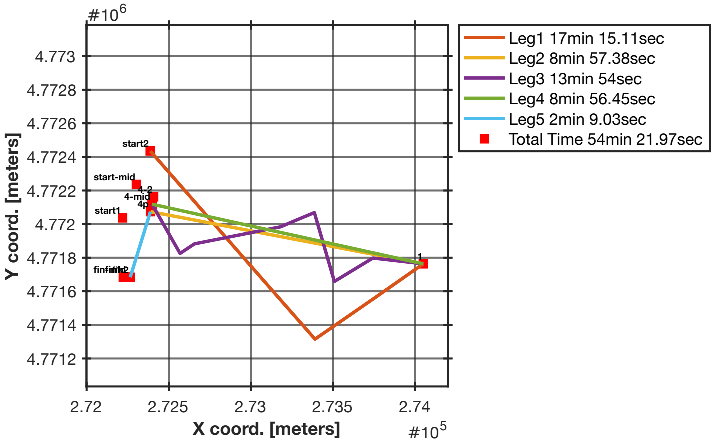
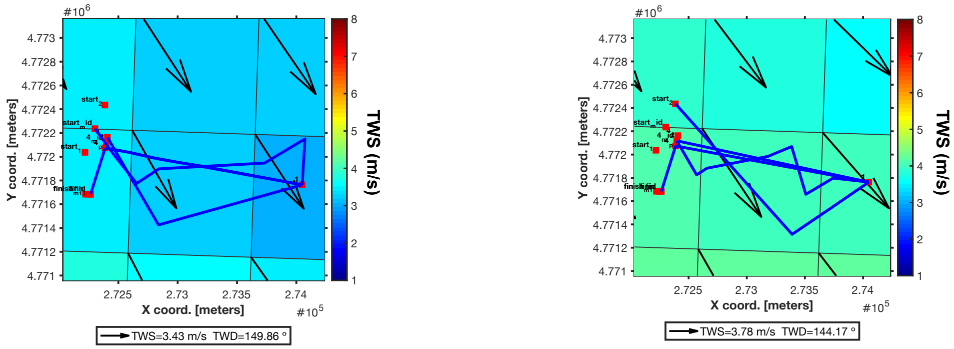


Figure 5.12: Optimal Time-Path Trajectories for the race using a Wind model constant for 60 (1 hr) minutes with grid size of 1 km. The time and leg trajectory are indicated by its color. The Optimal Time for this scenario is 54 min and 21.97 sec.

5.2.3. Wind field with a Time Step of 10 Minutes (1/6 Hour)

This scenario uses the WRF wind model as it is, where the wind field varies in space and time. The time step is 10 minutes, or 600 and a grid size about 1 km by side. As a result of this, the number of grid points increased significantly, and consequently the computational effort.

For the upwind legs, figure 5.14 shows that the top minimal trajectories for both legs



(a) Wind Field constant for 60 minutes and 1 km forecast conditions at 12:00 hrs over the racecourse.

(b) Wind Field constant for 60 minutes and 1 km forecast conditions at 13:00 hrs over the racecourse

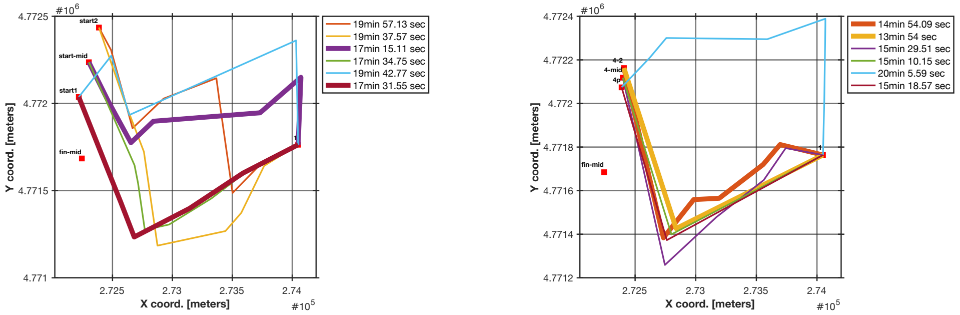
Figure 5.13: Minimal Time Path and Wind field conditions at the start and end of the race. This Wind Field is constant for 60 minutes (1 hr) with a grid size of 1 km.

follow to the south. For example, the start point of leg 1 is one in the middle and the other in the south. The path that starts at the middle point has more shifts in direction than the other path and it arrives in the next buoys with a final tack to the south. The other path which is the second-best initially goes to the south and then shifts direction to the north. The variation in time between them is only about 16 seconds.

For leg 3, the two minimal time trajectories start at the north point and both go to the south. The best of them have an "L" pattern instead of a zig-zag such as in the second best path. The difference in time between them is 1 minute and in this case, this difference is more related to the number of shifts. Moreover, most of the trajectories regardless where they start, they go to the south except for one which also has the largest time among them. This meand that the optimal area or direction for leg 3 is to go to the south.

The total time in this scenario is 51 minutes and 11.97 seconds until know it is the smallest time. The minimal time's trajectories for the upwind-mode starts at the mid and north point of the respective line. The leg 1 is the leg with the most tack maneuvers in contrast with the rest of the legs as a consequence its time is the largest. The rest of the legs do not follow properly a zig-zag pattern, leg 3 as mentioned before, follows an "L" shape with one tack maneuver, the rest of the legs follows a straight line. The optimal time/path is shown in figure 5.15. The red squares are the buoys and figure indicates the leg and its time with different colors.

Because more grid data points are available, the wind field variation is more accurately. For comparison purposes,figure 5.16 only shows 3 steps, these set of points are at 12:10 hrs, 12:20 hrs and 12:50 hrs. Comparing figure 5.16a with figure 5.16b the winds speed (V_{tw}) increases while the wind angle (β_{tw}) decreases.



(a) Minimal Time Paths for Leg 1 on upwind-mode for each of the star positions when the wind model is constant for 10 minutes with a space grid on 1 km.

(b) Minimal Time Paths for Leg 3 on upwind-mode for each of the star positions when the wind model is constant for 10 minutes with a space grid on 1 km

Figure 5.14: Upwind Legs 1 and 3 with Times per Leg and trajectories generated according to its position on the starting line and direction using a WRF wind model with a time step of 10 minutes and a grid size of 1 km

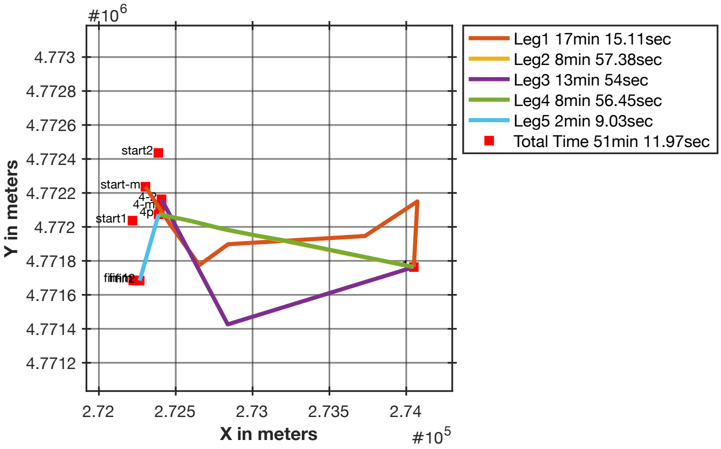
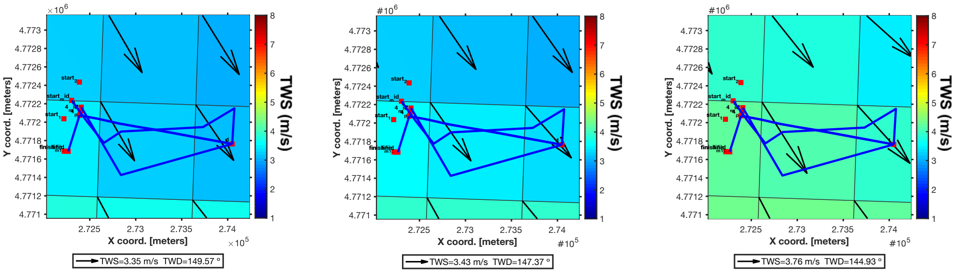


Figure 5.15: Optimal Time-Path using the WRF with a time step of 10 minutes and a grid size of 1 km. The trajectories and times of each leg are indicated by different colors.

5.2.4. WIND FIELD FROM THE RACE'S MEASUREMENTS

In this scenario, the wind data is from the race, where the sampling rate was 20 HZ and it comes from 5 locations dispersed around the courses areas. These measurements were at a height of 7.5 meters approx. so the algorithm converted to the CE height, similarly as in with previous scenarios.

The upwind legs for this scenarios show opposite directions to follow in contrast with previous scenarios particularly for the leg 1. Figure 5.17 also shows that the times are larger than the previous results. The preferable start direction remains to be the mid-point and the north point. The best times start at distinct points going to the north and



(a) Wind Field constant for 10 minutes and 1 km forecast conditions at 12:10 hrs over the racecourse.

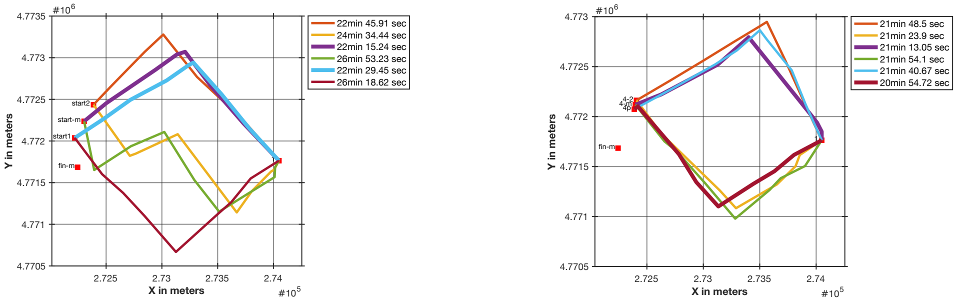
(b) Wind Field constant for 10 minutes and 1 km forecast conditions at 12:20 hrs over the racecourse.

(c) Wind Field constant for 10 minutes and 1 km forecast conditions at 12:50 hrs over the racecourse.

Figure 5.16: Minimal Time Path and Wind field conditions at three different times during the race. This Wind Field is constant for 10 minutes (1/6 hr) with a grid size of 1 km.

then do a significant shift to the heading direction to go to the buoy at its south. In fact, this last shift makes that both paths coincide until it arrives to the next buoy.

leg 3 has a similar pattern than previous sections an "L" or "V" shape with small tacking maneuvers in between. The best path starts at the south point while the second-best start at the middle point, the time difference between them is about 19 seconds. In this leg, all the half of the paths go to the north and the other to the south, in both cases all paths have a "V" shape.



(a) Minimal Time Paths for Leg 1 on upwind-mode for each of the star positions when the wind model uses the wind measurements taken at 20Hz. during the race.

(b) Minimal Time Paths for Leg 3 on upwind-mode for each of the star positions when the wind model uses the wind measurements taken at 20Hz. during the race

Figure 5.17: Upwind Legs 1 and 3 with Times per Leg and trajectories generated according to its position on the starting line and direction using the wind measurements taken during the race at 5 locations around the course at 20Hz.

Figure 5.18 shows the minimal time path for this scenario, the upwind sailing legs have a "V" shape while the rest of the legs sail in a straight line direction. The total time of the race is 63 minutes and 42.35 seconds, this time is much larger than previous results.

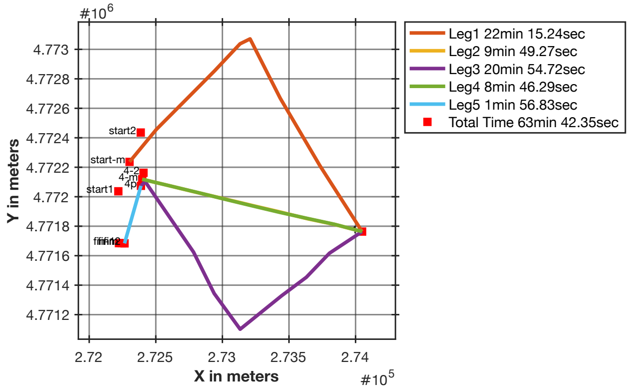
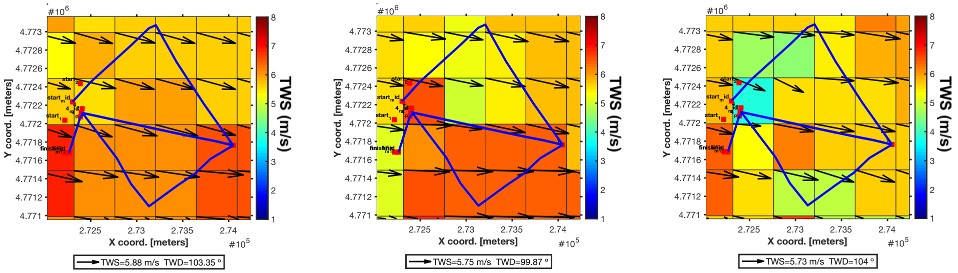


Figure 5.18: Optimal Time-Path using the wind measurements taken during the race at 5 locations around the race. The trajectories and times of each leg are indicated by different colors.

The wind field for this scenario does not have a grid size as the WRF wind model. The wind velocities are larger but with a wind direction smaller than the previous scenarios, this wind direction is close to 100°in contrast with the 144°from the WRF model. The variation of the wind at the start of the race at 12:10 hrs, 12:30hrs and 13:00 hrs are in figure 5.19.



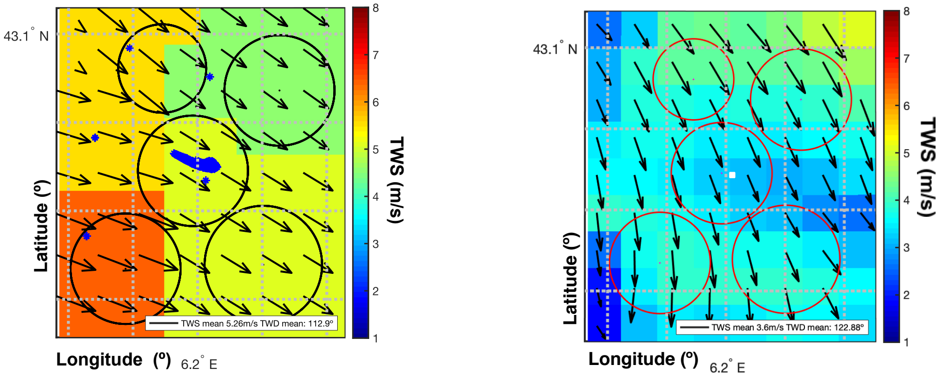
(a) Wind Field conditions at 12:10 hrs over the racecourse according to the wind measurements taken during the race.

(b) Wind Field conditions at 12:30 hrs over the racecourse according to the wind measurements taken during the race.

(c) Wind Field conditions at 13:00 hrs over the racecourse according to the wind measurements taken during the race.

Figure 5.19: Minimal Time Path and Wind field conditions at three different times during the race. This wind field was generated using the wind measurements taken during the race at 5 different locations around the race.

Moreover, the wind speed (V_{tw}) and direction(β_{tw}) over the racecourse R1 varies over time following a different pattern compared with the WRF wind model. For example, at the end of the race 13:00 hrs figure 5.19c and B.3c show the wind speed varies between 5 and 4 m/s. Figure 5.20a shows the locations of the race measurements with blue dots and its wind field properties. The spatial distribution of these locations are not uniform and the locations of many of the points for R1 and other courses are far from them. Meanwhile, many of the measurements are closer to the center of the *Echo* course thus, this algorithm estimated the wind properties of the race-lines via interpolation.Using the locations of the grid points from the WRF at 12:00 hrs figure 5.20 shows both wind fields. The wind speeds (V_{tw}) and the wind's angle (β_{tw}) of each is different furthermore, the wind properties of most of these points-locations results from an extrapolation method. This extrapolation especially influences the upwind modes, in other words, the conditions at which the wind sail at leg 1 and leg 3. These variations, particularly on the direction, are the reason why the paths have a different shape and form.



(a) Wind Field generated by the Wind Measurements taken during the race inside the area defined by the algorithm. This wind field uses the same locations as the WRF model to estimate the wind properties.

(b) WRF Wind Field and courses inside the area defined by the algorithm.

Figure 5.20: Wind Field forecast at 12:10 hrs. Comparison between wind velocities and direction from the WRF model and the wind measurements from the race. The blue dots on figure (a) are the locations from the measurements.

5.3. Comparison between the Results and the Race Winners

In this section, I compare the results of the previous scenarios with the results from the race. For this, only the top 10 winners are considered for the analysis. The *SAP Sailing Analytics*® website provides the information about to the times and paths. The comparison uses the times and paths. In the case of the times, the comparison uses the legs times and the race-time. The units used for the time are seconds and the order used is the same as the one from the scenario's section.

The summation of the duration of each leg results in the race-time. The collection of all the race times and legs got from the scenarios previously showed are in figure 5.21. In this figure, *Wind Measurements* refers to the results using the wind measurements from

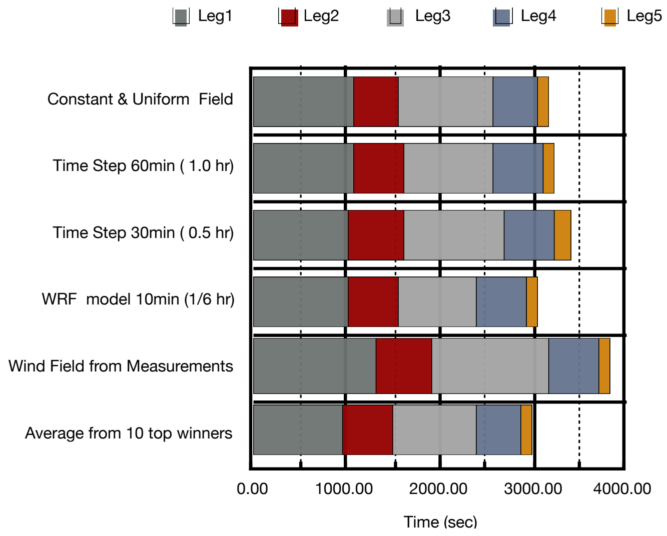
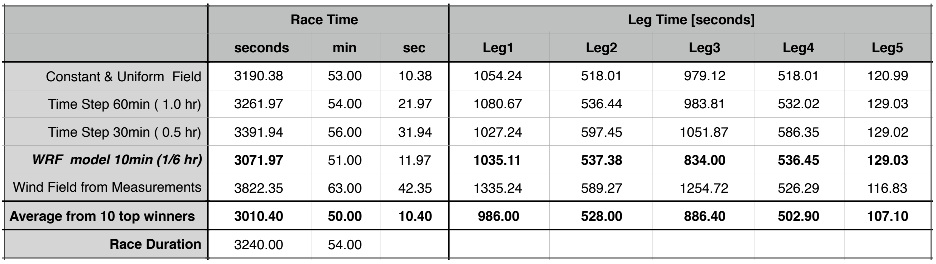


Figure 5.21: Times Legs by scenario and the average time of the top 10 winners for the race

the race. Comparing the legs times from each scenario against the average from the top 10 winners shows that the first leg except the wind measurements results is similar. Furthermore, leg 3 also sailing in upwind mode has a duration smaller than leg 1 and this condition only occurs in the WRF wind model.



	Race Time			Leg Time [seconds]				
	seconds	**min**	**sec**	**Leg1**	**Leg2**	**Leg3**	**Leg4**	**Leg5**
Constant & Uniform Field	3190.38	53.00	10.38	1054.24	518.01	979.12	518.01	120.99
Time Step 60min (1.0 hr)	3261.97	54.00	21.97	1080.67	536.44	983.81	532.02	129.03
Time Step 30min (0.5 hr)	3391.94	56.00	31.94	1027.24	597.45	1051.87	586.35	129.02
WRF model 10min (1/6 hr)	**3071.97**	51.00	11.97	**1035.11**	**537.38**	**834.00**	**536.45**	**129.03**
Wind Field from Measurements	3822.35	63.00	42.35	1335.24	589.27	1254.72	526.29	116.83
Average from 10 top winners	**3010.40**	**50.00**	**10.40**	**986.00**	**528.00**	**886.40**	**502.90**	**107.10**
Race Duration	3240.00	54.00						

Figure 5.22: Timess by Legs according the scenerios

When the leg times comparison is only for the WRF wind model, the average of the top 10 winners and the winner of the race. The leg 5 is significantly shorter in contrast with the resulted leg from the WRF wind model scenario, figure 5.23 shows this. Moreover, this leg according to the results from previous sections was sailing in a straight line, similarly happens to the leg 4.

The detailed comparison between the average results from the top 10 winners and the WRF wind model results are in figure 5.24. This shows that the race time difference between both results are close to 62 seconds, and this represents an error of -2% from

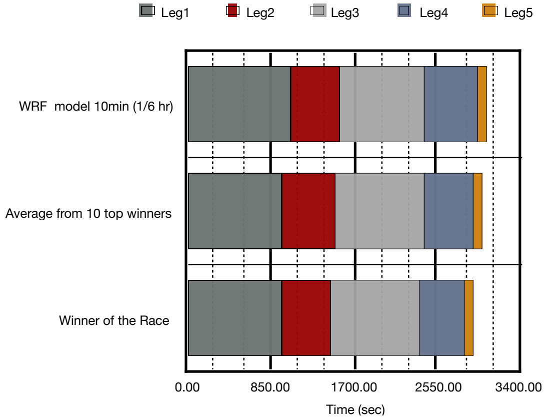
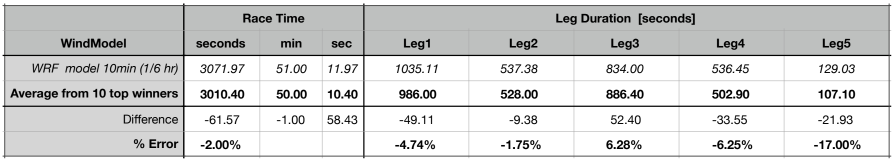


Figure 5.23: Leg's Times comparisons for the winners and the WRF wind field scenario.

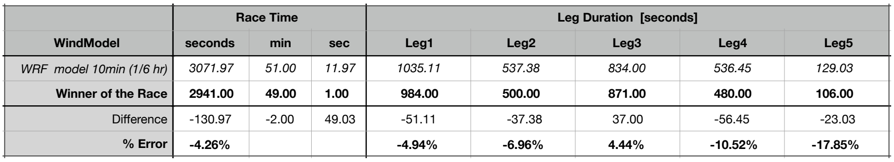
the average race result of the top 10 winners. Furthermore, the leg 5 has the largest percentage error which is 17% while the smallest error is for leg 2 with 1.75%. In fact, the rest of the legs have a percentage error between 4.75% and 6.28%.



	Race Time			Leg Duration [seconds]				
WindModel	seconds	min	sec	Leg1	Leg2	Leg3	Leg4	Leg5
WRF model 10min (1/6 hr)	*3071.97*	*51.00*	*11.97*	*1035.11*	*537.38*	*834.00*	*536.45*	*129.03*
Average from 10 top winners	**3010.40**	**50.00**	**10.40**	**986.00**	**528.00**	**886.40**	**502.90**	**107.10**
Difference	-61.57	-1.00	58.43	-49.11	-9.38	52.40	-33.55	-21.93
% Error	**-2.00%**			**-4.74%**	**-1.75%**	**6.28%**	**-6.25%**	**-17.00%**

Figure 5.24: Time Comparison by leg between the WRF wind results and the average of the top 10 winners.

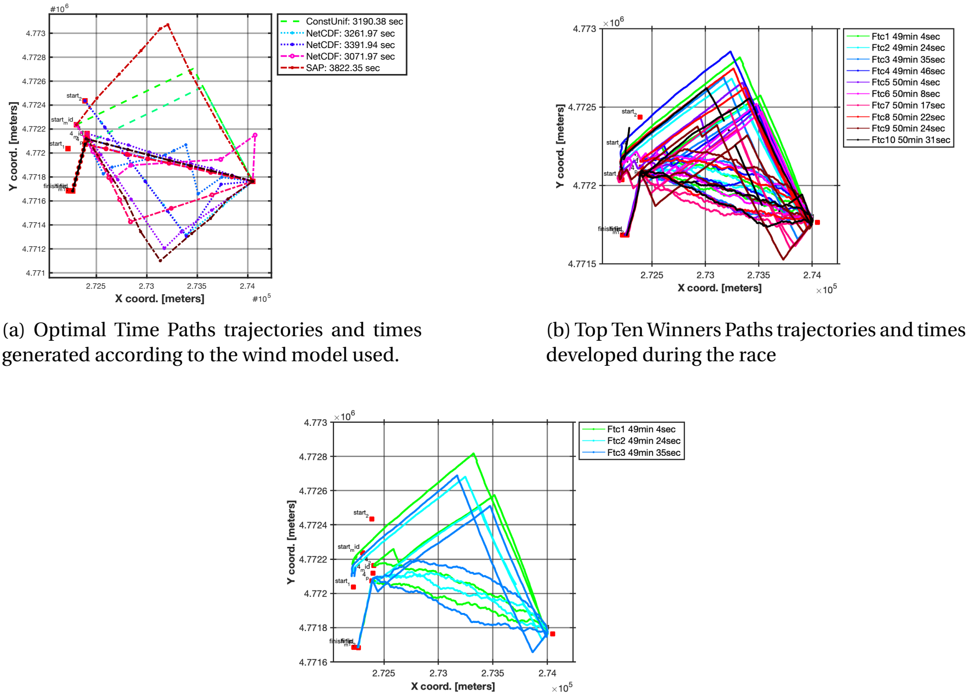
When the detailed comparison is between the WRF wind model and the winner of the race. The variations for the race time shows a time difference about 130.97 seconds which represents 4.26%, in fact, this error is double compared with the average of the top 10 winners of the race. The error for the leg 5 are still 17.85% additionally the error percentage for leg 4 is in this case 10.52% and for leg 2 it is 6.96%. The rest of the legs have an error of around 4% approx.



	Race Time			Leg Duration [seconds]				
WindModel	seconds	min	sec	Leg1	Leg2	Leg3	Leg4	Leg5
WRF model 10min (1/6 hr)	*3071.97*	*51.00*	*11.97*	*1035.11*	*537.38*	*834.00*	*536.45*	*129.03*
Winner of the Race	**2941.00**	**49.00**	**1.00**	**984.00**	**500.00**	**871.00**	**480.00**	**106.00**
Difference	-130.97	-2.00	49.03	-51.11	-37.38	37.00	-56.45	-23.03
% Error	**-4.26%**			**-4.94%**	**-6.96%**	**4.44%**	**-10.52%**	**-17.85%**

Figure 5.25: Time Comparison by leg between the WRF wind results and the winner of the race.

Another aspect to review after the times is the shape of the minimal time resulting from the scenarios and the developed paths from the race, these are in figure 5.26. For this, the comparison uses again the top ten competitors. The results from the top ten competitors show that the winners followed a north direction when the race starts for the upwind mode legs. In the other hand, for the downwind mode legs they do not follow a straight-line path as in the last case. Instead, they follow an alternative shape which neither looks like a zig-zag pattern. The simulation's results show for the upwind-mode paths that go to the north and south and straight lines paths for the downwind mode legs and the last leg.



(a) Optimal Time Paths trajectories and times generated according to the wind model used.

(b) Top Ten Winners Paths trajectories and times developed during the race

(c) Top Three Winners Paths trajectories and times developed during the race

Figure 5.26: Times and Paths generated by the wind models used and by the 10 top competitors of the race

If the comparison from the race only uses the top three winners as in figure 5.26c it shows that the downwind mode legs follow a path that has a shape of a curve. The winner of the race on the upwind mode legs has a larger length trajectory and for the downwind-mode, its trajectory is not a straight line. Besides this, the start points for the upwind-mode shows that for leg 1 the 3 winners starts around the midpoint. In contrast, for leg 3, the winner goes to the north point while the other two winners went to the south point. In these comparisons, it can see that the downwind mode legs follow path trajectories that contrast from the shaped developed by the scenarios.

To review the differences between the shapes from the scenarios and the shape of the winner, the paths are in the same plot. The first path scenario results from the constant and uniform wind field, figure 5.27 displays this. Leg 3, leg 4 and leg 5 coincide with the path followed by the winner. The leg 1 from the scenario goes to the north and it starts at the middle point similarly as the winner did. However, the figure shows that this leg does not coincide as much as the legs mentioned before, since only the last part of the leg has the same direction as the one from the winner.

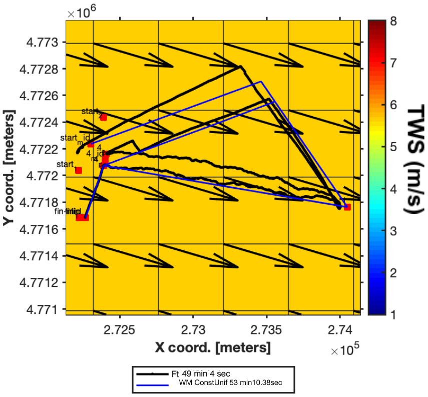


Figure 5.27: Optimal Time-Path indicated in blue using a Constant and Uniform Wind. The black line is the path developed by the winner of the race. The time difference between both paths is about 4 minutes

In the case of the path resulting from the WRF wind model, the legs do not coincide with the paths trajectories from the winner except for the last two legs, leg 4 and 5, this is displayed in figure 5.28. However, the start point for the upwind legs coincide. As mentioned before the time difference of the total race is only 130.97 seconds or 4.26 and despite that the leg 5 has the same path trajectory its times are different. Same happens with the leg 4 and in both cases, the error is about 17.85% and 10.53% respectively.

When the resulting race path from the wind measurements is compared with the path trajectory of the winner's race. Figure 5.29 displays that the last 2 legs coincide with the winner's path, this is not the case for the legs sailing under upwind-mode. Particularly, the leg 3 unlike the last two legs, this leg's path sails in the opposite direction than the winner, it sails to the south while the winner sails to the north. Leg 1, in both cases, has a "V" shape and it also starts at the middle point. Moreover, the trajectory after the first tack maneuver is similar. The time for the last two legs are 10.83 and 46.29 seconds for the leg 5 and leg 4 respectively, these differences represent approximately 10% from the winner's leg's times.

This section describes the results and how the time window and buoys locations from the race defines the parameters of the optimization algorithm. As a result of this, the algorithm solve the minimal time path for a variety of scenarios. Furthermore, using the winner's race times and path and contrasting with the results from the scenarios they show the differences and similarities between them. This comparison uses most of the time the average of the top ten winners. The comparison of the shape shows sometimes contrasting results. In addition, the evolution of the wind during the race shows the variations over the path development and how these influences its shape. The next section

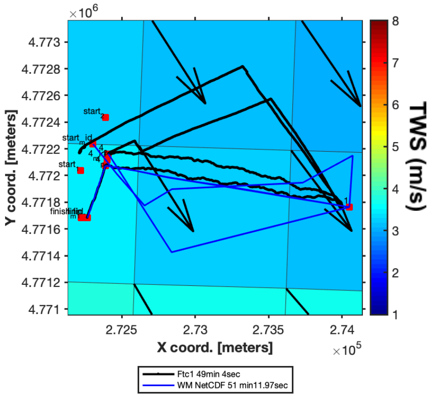


Figure 5.28: Optimal Time-Path indicated in blue using the WRF Wind Model (constant for 10 minutes). The black line is the path developed by the winner of the race. The time difference between both paths is about 2 minutes

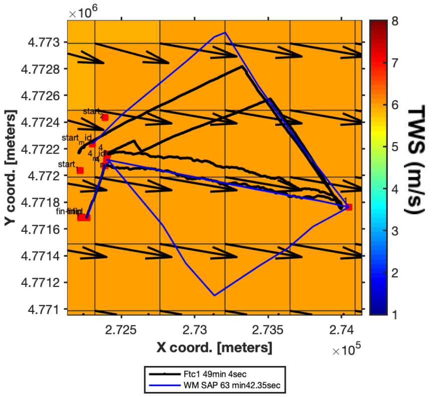


Figure 5.29: Optimal Time-Path indicated in blue using the wind measurements from the race. The black line is the path developed by the winner of the race.The time difference between both paths is about 14 minutes

is for the conclusion and recommendations of this research.

6

Conclusions and Recommendations

In conclusion, the minimal time path is conformed by the leg-times, start points over the race-lines and directions of the paths. Two of these elements were predicted similarly as the winners using the WRF wind model with an error in the race-time for less than 5%. However, the direction of the paths were not predicted accurately for the upwind-mode legs. This because the wind's direction predicted by the model had a difference about 10°from the direction reported during the race. This section will explain the conclusions reached by the results of the simulations and the recommendations for future work.

6.1. Conclusions

The minimal time path algorithm shaped by the wind for Olympic Classes, particularly for the Laser Class has shown sensibility to the scenarios delineated by alternative wind models. Despite the race-times and shapes differences derived from them, it shows that the WRF wind model with a grid resolution of 1 km and a time step of 10 minutes predict the same start point on all the race-lines as the winner of the race. The race-time using this model has an error approximately of 4%, this means that the model estimates a longer race time than the time made by the winner of the race. Even when this error is below 5% the direction of the paths for the upwind-mode legs is opposite to the direction taken by the top ten winners. In the other hand, the constant and uniform wind scenario had the same direction for the upwind-modes than the top ten winners.

The constant and uniform wind scenario predict the same direction for most of the legs as the top ten winners. The resulting race-time has an error about 8%. The start points were not the same for leg 2 and leg 3. Subsequently, the only leg direction that was not the same was leg 2.

Leg 2 is a downwind-mode leg and none of the scenarios predicted similar directions nor shapes as the top then winners. They sail this leg following shapes that looks like a curve/zig-zag instead of a straight-line. The reason for these shapes are not clear but it could be influenced by the current direction. Another consideration is the heavy traffic of dinghies especially because this leg is the second one.

The grid resolution of the wind model is important for the minimal time path estimation. More important is that the locations of the racecourse should be within the area limited by them this to avoid the extrapolation of the wind speed and angle at any point of the path. The extrapolation of the values such as the ones estimated by the wind measurements of the race rises errors (%) for leg-times sailed under upwind-mode larger than 25% from the leg-times of the winner. In fact, the wind measurements from the race shows that leg 3, sailed under upwind-mode, was in the opposite direction compared with the path of the top ten winners. If the wind field does not contain the race course then is more accurate to use the average values of both parameters and estimate the minimal path as it was under a constant and uniform wind field.

All the scenarios predict a straight-line trajectory for leg 5, the last leg, and the winners sailed it as predicted. Nonetheless, the leg-time in all the scenarios rise an error of about 17% for the WRF wind model or 11% for the constant and uniform wind field respect to the average leg-time from the top ten winners. This means that the speed of the boat and therefore the wind properties were not correctly. However, this difference accounts for the uncertainty of the speed of the Laser and the wind's speed in addition, to other sources of error.

The constant and uniform wind field differs from the WRF wind model in many aspects, however, the shape of the resulted minimal path of each compared with the paths of the winner's race reveals the effect of the wind direction on it. This difference is more significant than the variations on the race-time for each scenario. The speed of the wind affects how fast the dinghy arrives from one point to another, but it is the direction of the wind the parameter that defines the trajectories and shape of the time-path. For instance, the wind angle during the race was 108°, while the angle from the WRF wind model was 149.57°. The difference between them is sufficiently large to shape path trajectories with different directions.

6.2. Recommendations

Because of the larger errors in some of the leg-times, this research proposes to review the location of the CE. For the yachts, this is at 40% of the sail's height, on the Laser this distance could have additional consequences. Since the resulting locations are close to the sea level and the current and height of the waves could generate forces which are insignificant at higher distances.

The last recommendation is to model the sailboat in three-dimensions including waves and the current. Hence, the algorithm can quantify the influence of the position of the sail-man on the dinghy's speed particularly in the downwind and direct wind-mode where the optimal path is a straight-line. The current model only accounts for the displacement of the sail-man on the Y axis and not in the X axis which could be influenced by the height of the waves. This model does not considers this displacement and it assumes that the yaw moments are in balance always. The additional dimension to determine the location of the sail-man will influence the yaw and trimming moments and the center of gravity of the system. Because of this the forces and velocities for the boat could be different.

LIST OF FIGURES

References

[1] G. Sjøgaard, E. Inglés, and M. Narici, *Science in sailing: Interdisciplinary perspectives in optimizing sailing performance,* (2015).

[2] C. A. Marchaj, *Aero-hydrodynamics of sailing*, 3rd ed. (London (UK) Granada Pub., 1979).

[3] W. Sailing, *Rio 2016 sailing documentation,* (2016).

[4] *Race Management Policies for World Sailing Events (Fleet Racing)* (World Sailing, http://www.sailing.org/tools/documents/WorldSailingRaceManagementPoliciesJune2016-[20939].pdf, 2016).

[5] R. . O. C. for the Olympic and P. Games, eds., *Rio 2016 Olympic Sailing Competition. SAILING INSTRUCTIONS*, 2016 No. 20836, World Sailing (Rio 2016 Olympic Games Documentation, http://www.sailing.org/tools/documents/Rio2016SIs-[20836].pdf, 2016).

[6] M. Denny, *Float your boat!: the evolution and science of sailing* (JHU Press, 2009).

[7] T. T. Warner, *Numerical weather and climate prediction* (Cambridge University Press, 2010).

[8] A. Philpott and A. Mason, *Optimizing yacht routes under uncertainty,* in *The 15th Cheasapeake Sailing Yacht Symposium* (2001).

[9] MATLAB Optimization Toolbox, *Matlab optimization toolbox,* (2018 Student License), the MathWorks, Natick, MA, USA.

[10] *Laser radial characteristics and documentation,* (2016).

[11] A. Philpott, R. Sullivan, and P. Jackson, *Yacht velocity prediction using mathematical programming,* European Journal of Operational Research **67**, 13 (1993).

[12] C. S. Miller, *Sailing terms,* (2018).

[13] L. Larson and R. Eliasson, *Principles of Yacht Design. sl: Adlard Coles Nautical, 2007*, Tech. Rep. (ISBN 978-0-7136-7855-0, 2007).

[14] F. Fossati, *Aero-hydrodynamics and the performance of sailing yachts: the science behind sailing yachts and their design* (A&C Black, 2009).

[15] J. C. Alves and N. A. Cruz, *A mission programming system for an autonomous sailboat, 2014 Oceans - St. John's*, IEEE , 1 (2014).

[16] A. R. Claughton, J. F. Wellicome, and R. A. Shenoi, *Sailing yacht design: Practice*, Vol. 59 (Longman, 1998).

[17] T. Carrico, *A velocity prediction program for a planing dinghy,* THE 17th CHESAPEAKE SAILING YACHT SYMPOSIUM (2005).

[18] A. H. Day, *Performance prediction for sailing dinghies,* Ocean Engineering **136**, 67 (2017).

[19] C. Böhm, *A velocity prediction procedure for sailing yachts with a hydrodynamic model based on integrated fully coupled ranse-free-surface simulations,* (2014).

[20] J. H. Milgram, *Fluid mechanics for sailing vessel design,* Annual Review of Fluid Mechanics **30**, 613 (1998).

[21] B. Yang, L. Xiao, and J. Jouffroy, *A control-theoretic outlook at the no-go zone in sailing vessels,* in *OCEANS 2011* (IEEE, 2011) pp. 1–7.

[22] E. De Ridder, K. Vermeulen, and J. Keuning, *A mathematical model for the tacking maneuver of a sailing yacht,* in *Proceedings of the 18th International HISWA Symposium on Yacht Design and Construction, Amsterdam, Netherlands* (2004) pp. 1–34.

[23] J. Kimball, *Physics of sailing* (CRC Press, 2009).

[24] C. J. Patterson and J. D. Ridley, *Ship Stability, Powering and Resistance* (Adlard Coles Nautical, 2014).

[25] E. Spark and G. J. Connor, *Wind forecasting for the sailing events at the sydney 2000 olympic and paralympic games,* Weather and forecasting **19**, 181 (2004).

[26] C. Sheng, M. Xue, and S. Gao, *The structure and evolution of sea breezes during the qingdao olympics sailing test event in 2006,* Advances in Atmospheric Sciences **26**, 132 (2009).

[27] B. Golding, S. Ballard, K. Mylne, N. Roberts, A. Saulter, C. Wilson, P. Agnew, L. Davis, J. Trice, C. Jones, *et al.*, *Forecasting capabilities for the london 2012 olympics,* Bulletin of the American Meteorological Society **95**, 883 (2014).

[28] T. M. Giannaros, V. Kotroni, K. Lagouvardos, D. Dellis, P. Tsanakas, G. Mavrellis, P. Symeonidis, and T. Vakkas, *Ultrahigh resolution wind forecasting for the sailing events at the rio de janeiro 2016 summer olympic games,* Meteorological Applications **25**, 86 (2018).

[29] R. . O. C. for the Olympic and P. Games, eds., *Sailing Notice of Race,* 2016 No. 19974, World Sailing (Rio 2016 Olympic Games Documentation, http://www.sailing.org/tools/documents/SAILINGNoticeofRace-[19974].pdf, 2016).

[30] N. M. Kristensen, *Weather routing: Sensitivity to ensemble wind and current input.*, Master's thesis, UNIVERSITY OF OSLO Department of Geosciences-Meteorology and Oceanography section. (2010).

[31] T. Allsopp, A. Mason, and A. Philpott, *Optimal sailing routes with uncertain weather,* in *Proceedings of the 35th annual conference of the operational research society of New Zealand* (2000) pp. 65–74.

[32] T. Allsopp, *Stochastic weather routing for sailing vessels*, Ph.D. thesis, University of Auckland (1998).

[33] R. Rew and G. Davis, *Netcdf: an interface for scientific data access,* IEEE Computer Graphics and Applications **10**, 76 (1990).

[34] M. Pennanen *et al.*, *Optimal sailor position on an olympic dinghy*, Master's thesis, Aalto University, Departement of Applied Mechanics, Helsinki, Finland. (2015).

[35] J. S. Mitchell, *Geometric shortest paths and network optimization,* Handbook of computational geometry **334**, 633 (2000).

[36] I. S. Dolinskaya and R. L. Smith, *Fastest-path planning for direction-dependent speed functions,* Journal of Optimization Theory and Applications **158**, 480 (2013).

[37] M. P. Kelly, *Transcription methods for trajectory optimization,* Tutorial, Cornell University, Feb (2015).

[38] M. Rabaud, *Optimal routing in sailing,* (2016).

[39] M. Życzkowski, *Method of routing ships sailing in dedicated environment,* Annual of Navigation **24**, 147 (2017).

[40] H. Xing, X. Hu, G. Cao, and Y. Ge, *Path optimization algorithm for sailing along upwind-downwind course,* JOURNAL OF INFORMATION &COMPUTATIONAL SCIENCE **9**, 2141 (2012).

[41] S. Anarchy, *Yachts,* (2018).

[42] J. R. Binns, F. W. Bethwaite, and N. R. Saunders, *Development of a more realistic sailing simulator,* (2002).

[43] R. G. Flay, *A twisted flow wind tunnel for testing yacht sails,* Journal of Wind Engineering and Industrial Aerodynamics **63**, 171 (1996).

[44] *Laser Class Rules - One Design – International Laser Class Association,* (2015).

[45] R. van den Eijnde, *Trajectory optimization of sailing races*, Master's thesis, Delft University of Technology, Faculty of Aerospace Engineering (2012).

[46] L. Moreira and C. Guedes Soares, *Guidance and control of autonomous vehicles,* Marine Technology and Engineering. London, UK: Taylor & Francis Group , 503 (2011).

[47] J. Jouffroy, *On steering a sailing ship in a wearing maneuver,* IFAC Proceedings Volumes **42**, 26 (2009).

[48] Lin Xiao and J. Jouffroy, *Modeling and nonlinear heading control for sailing yachts,* in *OCEANS'11 MTS/IEEE KONA* (IEEE, 2011) pp. 1–6.

[49] I. S. Dolinskaya and A. Maggiar, *Time-optimal trajectories with bounded curvature in anisotropic media,* The International Journal of Robotics Research **31**, 1761 (2012).

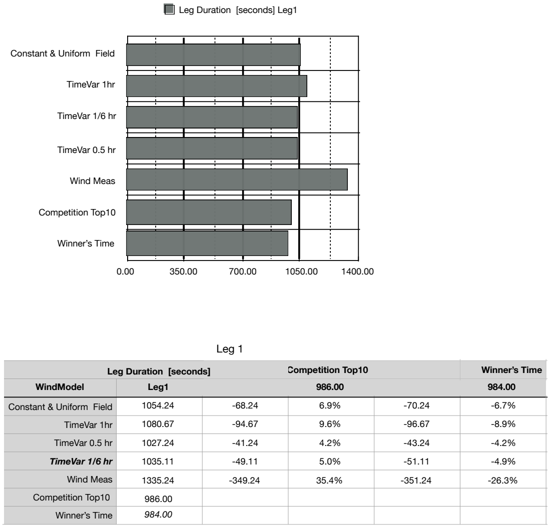
[50] *Race Management Policies for World Sailing Events (Fleet Racing)*, World Sailing, http://www.sailing.org/tools/documents/20171003Updatedto201720rulesandModifiedRaceM [23286].pdf, 2017th ed. (2017).

[51] *Sap sailing analytics- events-world cup series 2017 - hyères, france-laser,* (2016).

A

Leg Times Comparison by Wind Model

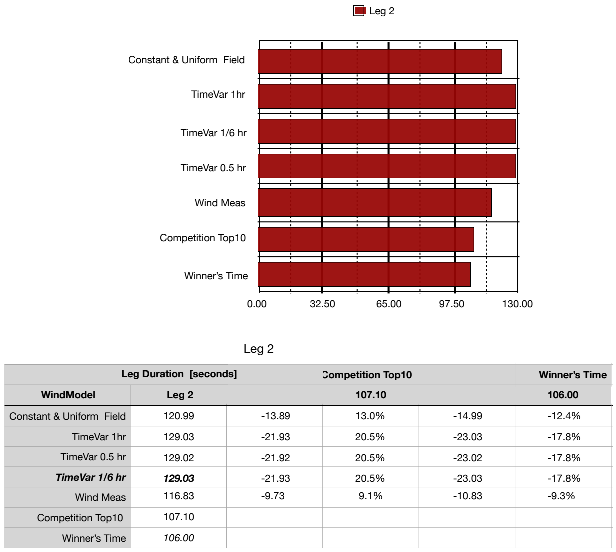
The next figures comperes the leg times according to the model used. The table shows the percentage difference against the top 10 winners and the winner of the race.



Leg 1

Leg Duration [seconds]		Competition Top10			Winner's Time
WindModel	**Leg1**	**986.00**			**984.00**
Constant & Uniform Field	1054.24	-68.24	6.9%	-70.24	-6.7%
TimeVar 1hr	1080.67	-94.67	9.6%	-96.67	-8.9%
TimeVar 0.5 hr	1027.24	-41.24	4.2%	-43.24	-4.2%
TimeVar 1/6 hr	1035.11	-49.11	5.0%	-51.11	-4.9%
Wind Meas	1335.24	-349.24	35.4%	-351.24	-26.3%
Competition Top10	986.00				
Winner's Time	*984.00*				

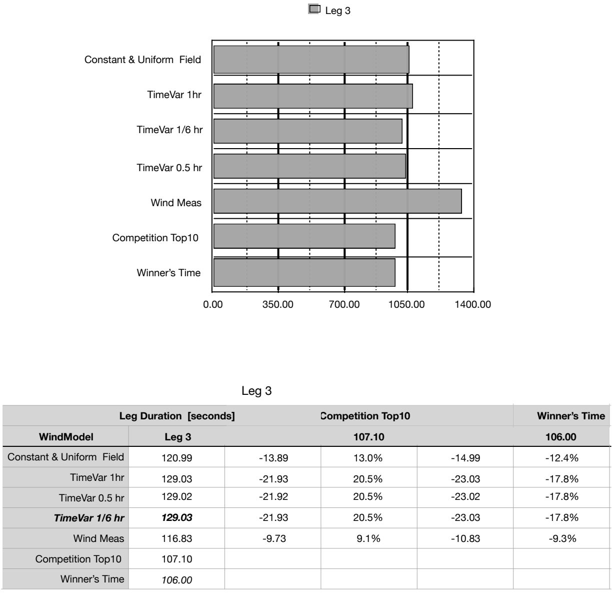
Figure A.1: Leg 1 Times by Wind Model



Leg 2

Leg Duration [seconds]		Competition Top10			Winner's Time
WindModel	**Leg 2**		**107.10**		**106.00**
Constant & Uniform Field	120.99	-13.89	13.0%	-14.99	-12.4%
TimeVar 1hr	129.03	-21.93	20.5%	-23.03	-17.8%
TimeVar 0.5 hr	129.02	-21.92	20.5%	-23.02	-17.8%
TimeVar 1/6 hr	***129.03***	-21.93	20.5%	-23.03	-17.8%
Wind Meas	116.83	-9.73	9.1%	-10.83	-9.3%
Competition Top10	107.10				
Winner's Time	*106.00*				

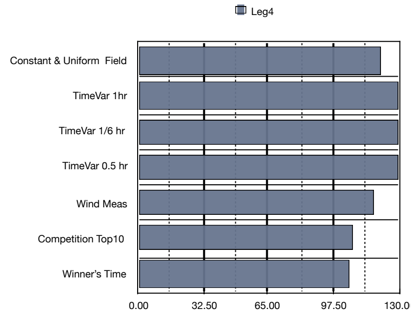
Figure A.2: Leg 2 Times by Wind Model



Leg 3

Leg Duration [seconds]		Competition Top10			Winner's Time
WindModel	**Leg 3**		**107.10**		**106.00**
Constant & Uniform Field	120.99	-13.89	13.0%	-14.99	-12.4%
TimeVar 1hr	129.03	-21.93	20.5%	-23.03	-17.8%
TimeVar 0.5 hr	129.02	-21.92	20.5%	-23.02	-17.8%
TimeVar 1/6 hr	***129.03***	-21.93	20.5%	-23.03	-17.8%
Wind Meas	116.83	-9.73	9.1%	-10.83	-9.3%
Competition Top10	107.10				
Winner's Time	*106.00*				

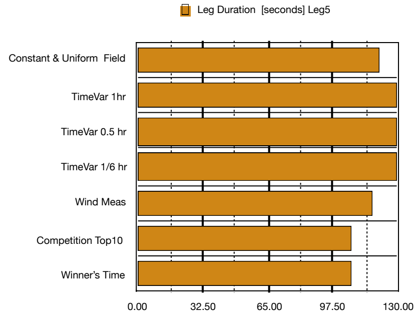
Figure A.3: Leg 3 Times by Wind Model



Leg 4

	Leg Duration [seconds]		Competition Top10		Winner's Time
WindModel	Leg4		107.10		106.00
Constant & Uniform Field	120.99	-13.89	13.0%	-14.99	-12.4%
TimeVar 1hr	129.03	-21.93	20.5%	-23.03	-17.8%
TimeVar 0.5 hr	129.02	-21.92	20.5%	-23.02	-17.8%
TimeVar 1/6 hr	***129.03***	-21.93	20.5%	-23.03	-17.8%
Wind Meas	116.83	-9.73	9.1%	-10.83	-9.3%
Competition Top10	107.10				
Winner's Time	*106.00*				

Figure A.4: Leg 4 Times by Wind Model



Leg 5

Results Comparison-2

	Leg Duration [seconds]		Competition Top10		Winner's Time
WindModel	Leg5		107.10		106.00
Constant & Uniform Field	120.99	-13.89	13.0%	-14.99	-12.4%
TimeVar 1hr	129.03	-21.93	20.5%	-23.03	-17.8%
TimeVar 0.5 hr	129.02	-21.92	20.5%	-23.02	-17.8%
TimeVar 1/6 hr	***129.03***	-21.93	20.5%	-23.03	-17.8%
Wind Meas	116.83	-9.73	9.1%	-10.83	-9.3%
Competition Top10	107.10				
Winner's Time	*106.00*				

Figure A.5: Leg 5 Times by Wind Model

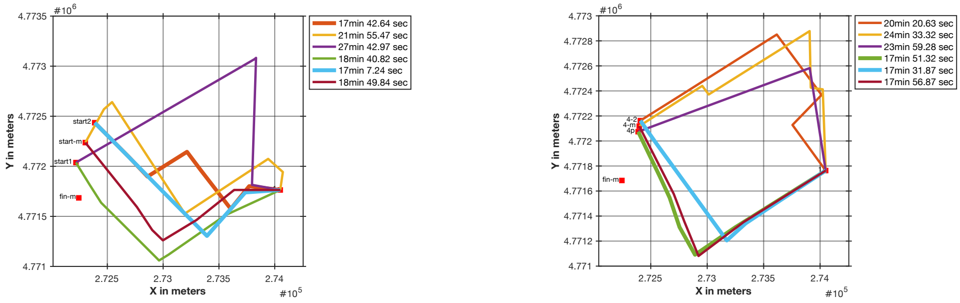
B

Additional Scenarios-Results

B.1. Wind Field with a Time Step of 30 Minutes (0.5 Hour)

The parameters of this scenario are wind constant for 0.5 hours or 30 minutes, and a grid of 1km per side. For the upwind mode the minimal time trajectories show similar directions than the previous scenarios. The best paths of the leg 1 goes to the south, regardless of its start locations point, only one path goes to the north. Furthermore, the top times for the leg 1 start at the North point of the start line and the trajectories overlap in many sections, see figure B.1a. As a consequence of this, the times differ only by 30 seconds approx, and both trajectories show a zig-zag pattern with at least 3 tacks maneuvers.

Leg 3 on figure B.1b shows it has a similar shape than leg 1, here the start point of the two minimal time trajectories are opposite, one starts at the North point while the other is at the South point, however in both cases, they go to the south.



(a) Leg 1 trajectories

(b) Leg 3 trajectories

Figure B.1: Upwind Legs Times from using a wind field constant for 0.5hr

The total time of the race for this scenario is 56 minutes with 31.94 seconds. The times of the leg 1 and 3 are similar, 17 minutes with 7.24 seconds and 17 minutes with

31.87 seconds. Even when the distance of the leg 3 is smaller than the leg 1, it has a longer time trajectory, B.2. Furthermore, on leg 3, the number of maneuvers is less than at leg 1 and despite this, its time is longer. The downwind mode's legs show trajectories close to a straight line with tack maneuvers smaller than in the upwind mode.

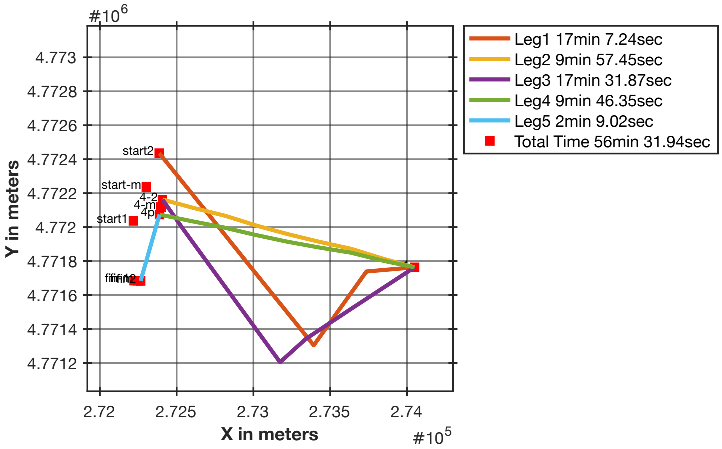
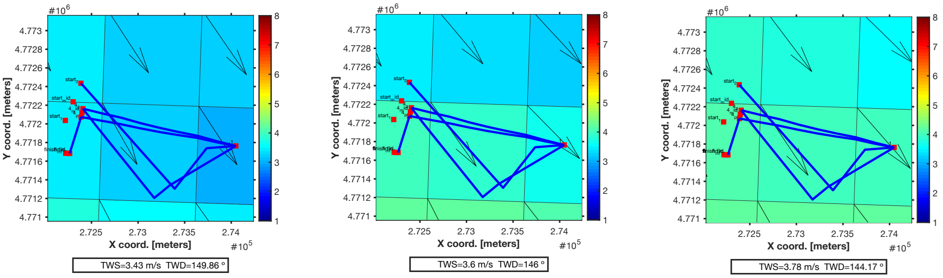


Figure B.2: Total Times and Leg's time for a wind field constant for 0.5 hr

The wind properties used on this scenario are from the set points at 12:00 hrs, 12:30 hrs and 13:00 hrs. The wind field pattern at these times is in figure B.3. The figure B.3b shows the transition between speeds, the top side has lower velocities than the bottom side. The wind shift directions between the three of them are approx. 2 °. The resultant trajectory of this scenario and the previous one are different, not only in time but in shape also. In this scenario, the number of tacks is less than with a wind constant for one hour however the time is longer than it. For the leg 3 even when the wind is stronger, the variation on its angle is larger than at the end.

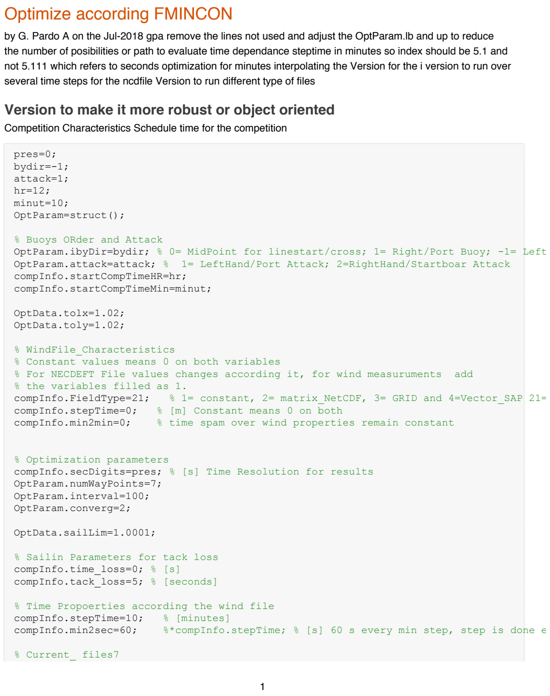


(a) Wind Field at 12:00 hrs (b) Wind Field at 12:30 hrs (c) Wind Field at 13:00 hrs.

Figure B.3: Minimal Time Path generated by a Wind Field updated every 30 minutes (0.5 hr).

C

ALGORITHM



Optimize according FMINCON

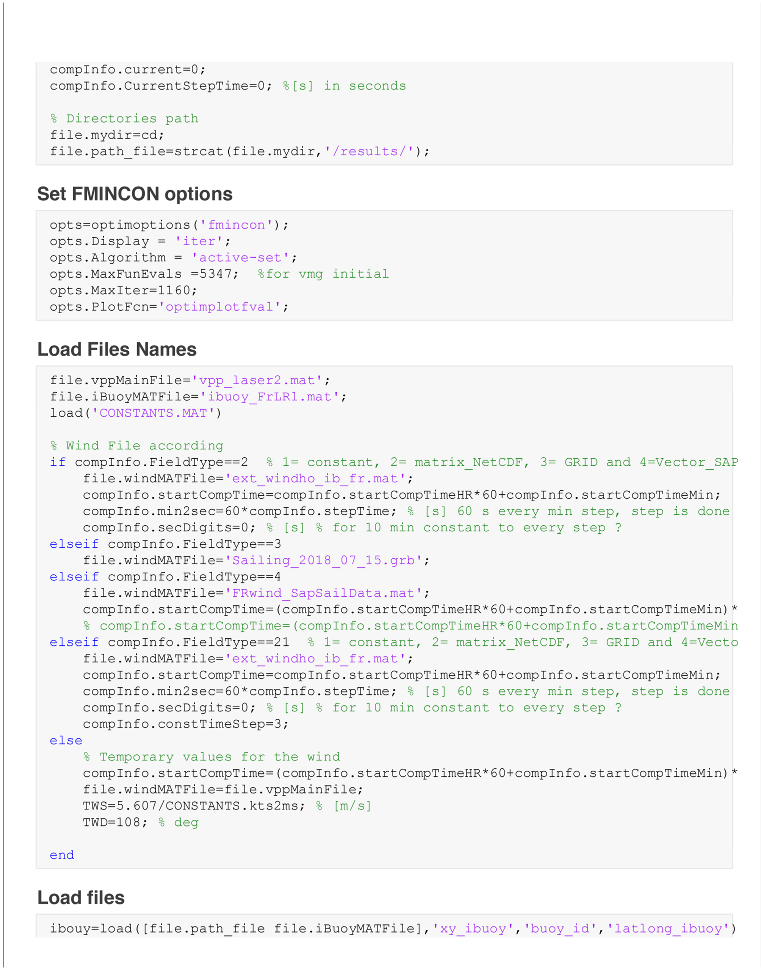
by G. Pardo A on the Jul-2018 gpa remove the lines not used and adjust the OptParam.lb and up to reduce the number of posibilities or path to evaluate time dependance steptime in minutes so index should be 5.1 and not 5.111 which refers to seconds optimization for minutes interpolating the Version for the i version to run over several time steps for the ncdfile Version to run different type of files

Version to make it more robust or object oriented

Competition Characteristics Schedule time for the competition

```
pres=0;
bydir=-1;
attack=1;
hr=12;
minut=10;
OptParam=struct();

% Buoys ORder and Attack
OptParam.ibyDir=bydir; % 0= MidPoint for linestart/cross; 1= Right/Port Buoy; -1= Left
OptParam.attack=attack; %  1= LeftHand/Port Attack; 2=RightHand/Startboar Attack
compInfo.startCompTimeHR=hr;
compInfo.startCompTimeMin=minut;

OptData.tolx=1.02;
OptData.toly=1.02;

% WindFile_Characteristics
% Constant values means 0 on both variables
% For NECDEFT File values changes according it, for wind measurments  add
% the variables filled as 1.
compInfo.FieldType=21;   % 1= constant, 2= matrix_NetCDF, 3= GRID and 4=Vector_SAP 21=
compInfo.stepTime=0;   % [m] Constant means 0 on both
compInfo.min2min=0;    % time spam over wind properties remain constant


% Optimization parameters
compInfo.secDigits=pres; % [s] Time Resolution for results
OptParam.numWayPoints=7;
OptParam.interval=100;
OptParam.converg=2;

OptData.sailLim=1.0001;

% Sailin Parameters for tack loss
compInfo.time_loss=0; % [s]
compInfo.tack_loss=5; % [seconds]

% Time Propoerties according the wind file
compInfo.stepTime=10;   % [minutes]
compInfo.min2sec=60;    %*compInfo.stepTime; % [s] 60 s every min step, step is done e

% Current_ files7
```

1



```
compInfo.current=0;
compInfo.CurrentStepTime=0; %[s] in seconds

% Directories path
file.mydir=cd;
file.path_file=strcat(file.mydir,'/results/');
```

Set FMINCON options

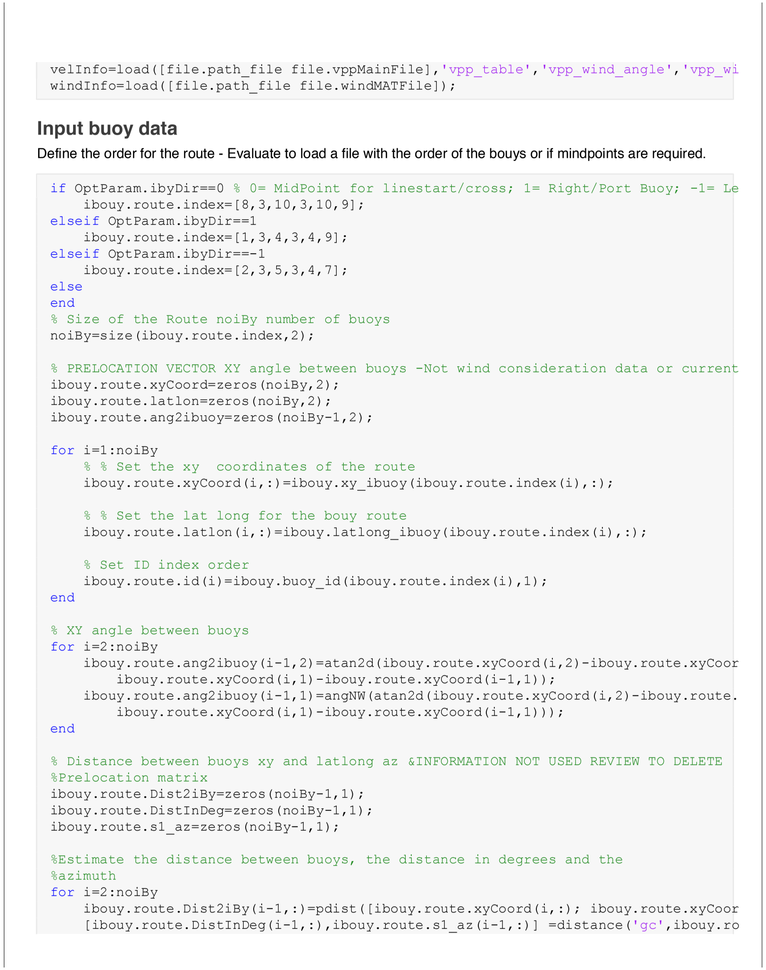
```
opts=optimoptions('fmincon');
opts.Display = 'iter';
opts.Algorithm = 'active-set';
opts.MaxFunEvals =5347;  %for vmg initial
opts.MaxIter=1160;
opts.PlotFcn='optimplotfval';
```

Load Files Names

```
file.vppMainFile='vpp_laser2.mat';
file.iBuoyMATFile='ibuoy_FrLR1.mat';
load('CONSTANTS.MAT')

% Wind File according
if compInfo.FieldType==2  % 1= constant, 2= matrix_NetCDF, 3= GRID and 4=Vector_SAP
    file.windMATFile='ext_windho_ib_fr.mat';
    compInfo.startCompTime=compInfo.startCompTimeHR*60+compInfo.startCompTimeMin;
    compInfo.min2sec=60*compInfo.stepTime; % [s] 60 s every min step, step is done
    compInfo.secDigits=0; % [s] % for 10 min constant to every step ?
elseif compInfo.FieldType==3
    file.windMATFile='Sailing_2018_07_15.grb';
elseif compInfo.FieldType==4
    file.windMATFile='FRwind_SapSailData.mat';
    compInfo.startCompTime=(compInfo.startCompTimeHR*60+compInfo.startCompTimeMin)*
    % compInfo.startCompTime=(compInfo.startCompTimeHR*60+compInfo.startCompTimeMin
elseif compInfo.FieldType==21  % 1= constant, 2= matrix_NetCDF, 3= GRID and 4=Vecto
    file.windMATFile='ext_windho_ib_fr.mat';
    compInfo.startCompTime=compInfo.startCompTimeHR*60+compInfo.startCompTimeMin;
    compInfo.min2sec=60*compInfo.stepTime; % [s] 60 s every min step, step is done
    compInfo.secDigits=0; % [s] % for 10 min constant to every step ?
    compInfo.constTimeStep=3;
else
    % Temporary values for the wind
    compInfo.startCompTime=(compInfo.startCompTimeHR*60+compInfo.startCompTimeMin)*
    file.windMATFile=file.vppMainFile;
    TWS=5.607/CONSTANTS.kts2ms; % [m/s]
    TWD=108; % deg

end
```

Load files

```
ibouy=load([file.path_file file.iBuoyMATFile],'xy_ibuoy','buoy_id','latlong_ibuoy')
```



```
velInfo=load([file.path_file file.vppMainFile],'vpp_table','vpp_wind_angle','vpp_wi
windInfo=load([file.path_file file.windMATFile]);
```

Input buoy data

Define the order for the route - Evaluate to load a file with the order of the bouys or if mindpoints are required.

```
if OptParam.ibyDir==0 % 0= MidPoint for linestart/cross; 1= Right/Port Buoy; -1= Le
    ibouy.route.index=[8,3,10,3,10,9];
elseif OptParam.ibyDir==1
    ibouy.route.index=[1,3,4,3,4,9];
elseif OptParam.ibyDir==-1
    ibouy.route.index=[2,3,5,3,4,7];
else
end
% Size of the Route noiBy number of buoys
noiBy=size(ibouy.route.index,2);

% PRELOCATION VECTOR XY angle between buoys -Not wind consideration data or current
ibouy.route.xyCoord=zeros(noiBy,2);
ibouy.route.latlon=zeros(noiBy,2);
ibouy.route.ang2ibuoy=zeros(noiBy-1,2);

for i=1:noiBy
    % % Set the xy  coordinates of the route
    ibouy.route.xyCoord(i,:)=ibouy.xy_ibuoy(ibouy.route.index(i),:);

    % % Set the lat long for the bouy route
    ibouy.route.latlon(i,:)=ibouy.latlong_ibuoy(ibouy.route.index(i),:);

    % Set ID index order
    ibouy.route.id(i)=ibouy.buoy_id(ibouy.route.index(i),1);
end

% XY angle between buoys
for i=2:noiBy
    ibouy.route.ang2ibuoy(i-1,2)=atan2d(ibouy.route.xyCoord(i,2)-ibouy.route.xyCoor
        ibouy.route.xyCoord(i,1)-ibouy.route.xyCoord(i-1,1));
    ibouy.route.ang2ibuoy(i-1,1)=angNW(atan2d(ibouy.route.xyCoord(i,2)-ibouy.route.
        ibouy.route.xyCoord(i,1)-ibouy.route.xyCoord(i-1,1)));
end

% Distance between buoys xy and latlong az &INFORMATION NOT USED REVIEW TO DELETE
%Prelocation matrix
ibouy.route.Dist2iBy=zeros(noiBy-1,1);
ibouy.route.DistInDeg=zeros(noiBy-1,1);
ibouy.route.s1_az=zeros(noiBy-1,1);

%Estimate the distance between buoys, the distance in degrees and the
%azimuth
for i=2:noiBy
    ibouy.route.Dist2iBy(i-1,:)=pdist([ibouy.route.xyCoord(i,:); ibouy.route.xyCoor
    [ibouy.route.DistInDeg(i-1,:),ibouy.route.s1_az(i-1,:)] =distance('gc',ibouy.ro
end
```

```
ibouy.route.StartL=pdist([ibouy.xy_ibuoy(1,:); ibouy.xy_ibuoy(2,:)]);
[ibouy.route.StartL2,ibouy.route.sL_az] =distance('gc',ibouy.latlong_ibuoy(1,:),ibo

ibouy.route.EndL=pdist([ibouy.xy_ibuoy(6,:); ibouy.xy_ibuoy(7,:)]);
[ibouy.route.EndL2,ibouy.route.eL_az] =distance('gc',ibouy.latlong_ibuoy(6,:),ibouy

ibouy.route.L4=pdist([ibouy.xy_ibuoy(4,:); ibouy.xy_ibuoy(5,:)]);
[ibouy.route.L42,ibouy.route.L4_az] =distance('gc',ibouy.latlong_ibuoy(4,:),ibouy.l
```

Bounds

```
ibouy.route.xyCoordCTRpt=mean(ibouy.route.xyCoord);
% % OptParam.YminTemp=ibouy.route.xyCoordCTRpt(:,2)+min(ibouy.xy_ibuoy(:,2))/OptPar
```

Optimization Optimization an optimal path using FMINCON

```
iphase=1;
t0min=0; % [s] seconds since the velocity is given in meters per second, time at th
tfmin=0; % final time for the leg.
compInfo.tfmin(iphase)=0;
compInfo.t0min(iphase)=0;
```

Wind field

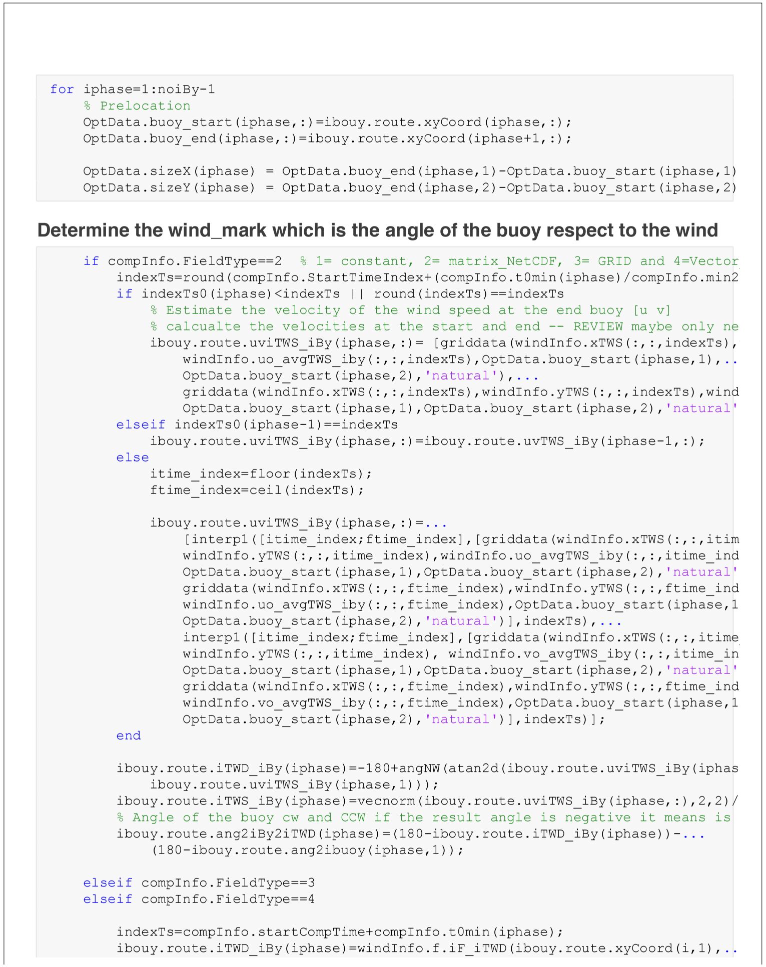
```
if compInfo.FieldType==2  % 1= constant, 2= matrix_NetCDF, 3= GRID and 4=Vector_SAP
    windInfo.uo_avgTWS_iby= double(squeeze(windInfo.u_avgTWS_iby(:,:,1,:)));
    windInfo.vo_avgTWS_iby=double(squeeze(windInfo.v_avgTWS_iby(:,:,1,:)));
    [windInfo.x_ll,windInfo.y_ll,~]=deg2utm(reshape(windInfo.xlat_tws_ibouy, [],1),
        reshape(windInfo.xlong_tws_ibouy, [],1));

    windInfo.xTWS=reshape(windInfo.x_ll,size(windInfo.xlong_tws_ibouy,1),size(windI
        size(windInfo.xlong_tws_ibouy,3));
    windInfo.yTWS=reshape(windInfo.y_ll,size(windInfo.xlat_tws_ibouy,1),size(windIn
        size(windInfo.xlat_tws_ibouy,3));

    if floor(compInfo.startCompTimeMin/compInfo.stepTime)*compInfo.stepTime==floor(
        compInfo.StartTimeID=(compInfo.startCompTimeHR*60/compInfo.stepTime+...
            compInfo.startCompTimeMin/compInfo.stepTime)+1;
        compInfo.StartTimeIndex=windInfo.itime(compInfo.StartTimeID,5);
    else
        compInfo.StartTimeID(1)=(compInfo.startCompTimeHR*60/compInfo.stepTime+...
            floor(compInfo.startCompTimeMin/compInfo.stepTime))+1;
        compInfo.StartTimeID(2)=(compInfo.startCompTimeHR*60/compInfo.stepTime+...
            ceil(compInfo.startCompTimeMin/compInfo.stepTime))+1;
        compInfo.StartTimeIndex(1)=windInfo.itime(compInfo.StartTimeID(1),5);
        compInfo.StartTimeIndex(2)=windInfo.itime(compInfo.StartTimeID(2),5);
    end

    indexTs0=round((compInfo.StartTimeIndex(1)+compInfo.t0min(1)/compInfo.min2sec),
```

```
elseif compInfo.FieldType==3
elseif compInfo.FieldType==4  % 1= constant, 2= matrix_NetCDF, 3= GRID and 4=Vector
    windInfo.x_ll=windInfo.f.ix;
    windInfo.y_ll=windInfo.f.iy;
    windInfo.xTWS=sind(windInfo.f.id).*windInfo.f.iv;
    windInfo.yTWS=cosd(windInfo.f.id).*windInfo.f.iv;
    IndexTs0=compInfo.startCompTime+compInfo.t0min(1);
elseif compInfo.FieldType==1
    windInfo.TWS=TWS;
    windInfo.TWD=TWD;
    IndexTs0=compInfo.startCompTime+compInfo.t0min(1);
elseif compInfo.FieldType==21  % 1= constant, 2= matrix_NetCDF, 3= GRID and 4=Vecto

    windInfo.xlat0_tws_ibouy=windInfo.xlat_tws_ibouy;
    windInfo.xlong0_tws_ibouy=windInfo.xlong_tws_ibouy;
    windInfo = rmfield(windInfo,'xlat_tws_ibouy');
    windInfo = rmfield(windInfo,'xlong_tws_ibouy');
    idx=windInfo.itime(find(windInfo.itime(:,3)==compInfo.startCompTimeHR,1),5);

    for i=-(6/compInfo.constTimeStep):2
        % idx2= idx+(i+compInfo.sonsTimeStep);
        windInfo.uo_avgTWS_iby(:,:,i+3)= double(squeeze(windInfo.u_avgTWS_iby(:,:,1
        windInfo.vo_avgTWS_iby(:,:,i+3)= double(squeeze(windInfo.v_avgTWS_iby(:,:,1
        windInfo.xlat_tws_ibouy(:,:,i+3)=double(squeeze(windInfo.xlat0_tws_ibouy(:,
        windInfo.xlong_tws_ibouy(:,:,i+3)=double(squeeze(windInfo.xlong0_tws_ibouy(
    end

    [windInfo.x_ll,windInfo.y_ll,~]=deg2utm(reshape(windInfo.xlat_tws_ibouy, [],1),
        reshape(windInfo.xlong_tws_ibouy, [],1));

    windInfo.xTWS=reshape(windInfo.x_ll,size(windInfo.xlong_tws_ibouy,1),size(windI
        size(windInfo.xlong_tws_ibouy,3));
    windInfo.yTWS=reshape(windInfo.y_ll,size(windInfo.xlat_tws_ibouy,1),size(windIn
        size(windInfo.xlat_tws_ibouy,3));

    indexTs0_refMIN=compInfo.startCompTime+compInfo.t0min(1);
    compInfo.startCompTimeIdx=6/compInfo.constTimeStep+1; % 1 hr has 6 steps of ten
    indexTs0=compInfo.startCompTimeIdx;
end
```

SETUp for location of bouys % REVIEW 'TWD_polar_rel'.

```
InfoLink(iphase,:)=struct('xy_coord',[0,0],'time',[0],'TWS',[0],'TWD',[0,0],'TWD_ib
WayPoints=struct();
link_cond(iphase)="NA";
buoy_link=zeros(1,2);
PointsAlt=struct();
leg_mtp(iphase)=struct('time',[],'x',[],'y',[],'id',"",'sec2min',[],'secdec',[]);
legID={'StraightLine','w1Attack','w1Attack2','optSL','optVMG','optVMG2'};
ibouy.route.uvTWS_iBy=zeros(noiBy-1,2);
```

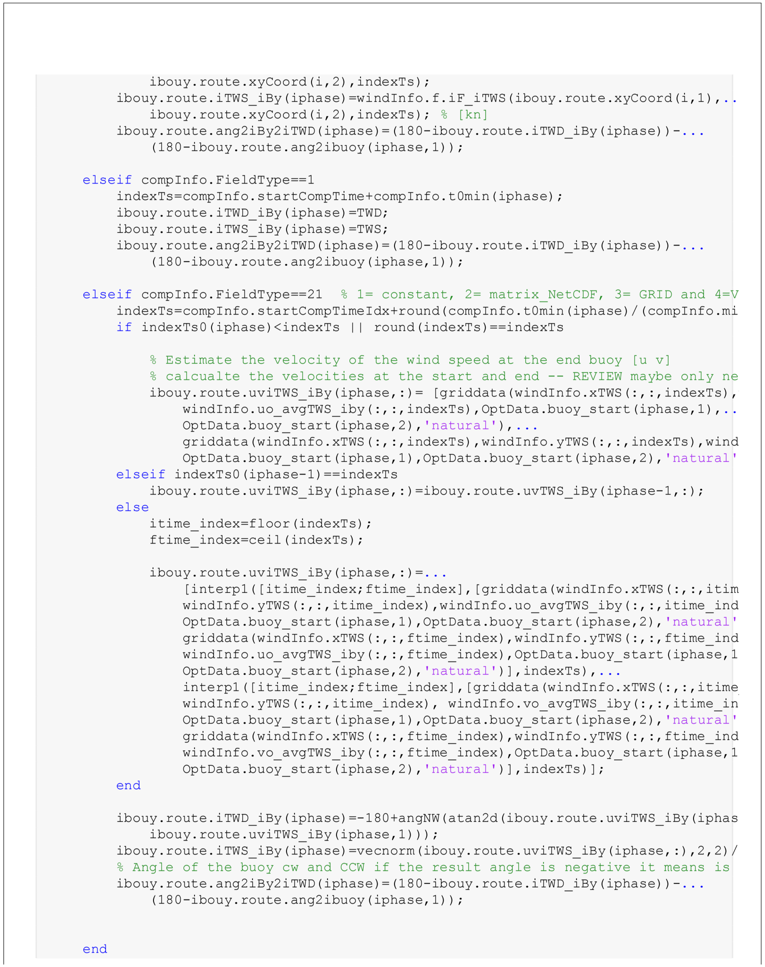
FMINCON



```
for iphase=1:noiBy-1
    % Prelocation
    OptData.buoy_start(iphase,:)=ibouy.route.xyCoord(iphase,:);
    OptData.buoy_end(iphase,:)=ibouy.route.xyCoord(iphase+1,:);

    OptData.sizeX(iphase) = OptData.buoy_end(iphase,1)-OptData.buoy_start(iphase,1)
    OptData.sizeY(iphase) = OptData.buoy_end(iphase,2)-OptData.buoy_start(iphase,2)
```

Determine the wind_mark which is the angle of the buoy respect to the wind

```
    if compInfo.FieldType==2  % 1= constant, 2= matrix_NetCDF, 3= GRID and 4=Vector
        indexTs=round(compInfo.StartTimeIndex+(compInfo.t0min(iphase)/compInfo.min2
        if indexTs0(iphase)<indexTs || round(indexTs)==indexTs
            % Estimate the velocity of the wind speed at the end buoy [u v]
            % calcualte the velocities at the start and end -- REVIEW maybe only ne
            ibouy.route.uviTWS_iBy(iphase,:)= [griddata(windInfo.xTWS(:,:,indexTs),
                windInfo.uo_avgTWS_iby(:,:,indexTs),OptData.buoy_start(iphase,1),..
                OptData.buoy_start(iphase,2),'natural'),...
                griddata(windInfo.xTWS(:,:,indexTs),windInfo.yTWS(:,:,indexTs),wind
                OptData.buoy_start(iphase,1),OptData.buoy_start(iphase,2),'natural'
        elseif indexTs0(iphase-1)==indexTs
            ibouy.route.uviTWS_iBy(iphase,:)=ibouy.route.uvTWS_iBy(iphase-1,:);
        else
            itime_index=floor(indexTs);
            ftime_index=ceil(indexTs);

            ibouy.route.uviTWS_iBy(iphase,:)=...
                [interp1([itime_index;ftime_index],[griddata(windInfo.xTWS(:,:,itim
                windInfo.yTWS(:,:,itime_index),windInfo.uo_avgTWS_iby(:,:,itime_ind
                OptData.buoy_start(iphase,1),OptData.buoy_start(iphase,2),'natural'
                griddata(windInfo.xTWS(:,:,ftime_index),windInfo.yTWS(:,:,ftime_ind
                windInfo.uo_avgTWS_iby(:,:,ftime_index),OptData.buoy_start(iphase,1
                OptData.buoy_start(iphase,2),'natural')],indexTs),...
                interp1([itime_index;ftime_index],[griddata(windInfo.xTWS(:,:,itime
                windInfo.yTWS(:,:,itime_index), windInfo.vo_avgTWS_iby(:,:,itime_in
                OptData.buoy_start(iphase,1),OptData.buoy_start(iphase,2),'natural'
                griddata(windInfo.xTWS(:,:,ftime_index),windInfo.yTWS(:,:,ftime_ind
                windInfo.vo_avgTWS_iby(:,:,ftime_index),OptData.buoy_start(iphase,1
                OptData.buoy_start(iphase,2),'natural')],indexTs)];
        end

        ibouy.route.iTWD_iBy(iphase)=-180+angNW(atan2d(ibouy.route.uviTWS_iBy(iphas
            ibouy.route.uviTWS_iBy(iphase,1)));
        ibouy.route.iTWS_iBy(iphase)=vecnorm(ibouy.route.uviTWS_iBy(iphase,:),2,2)/
        % Angle of the buoy cw and CCW if the result angle is negative it means is
        ibouy.route.ang2iBy2iTWD(iphase)=(180-ibouy.route.iTWD_iBy(iphase))-...
            (180-ibouy.route.ang2ibuoy(iphase,1));

    elseif compInfo.FieldType==3
    elseif compInfo.FieldType==4

        indexTs=compInfo.startCompTime+compInfo.t0min(iphase);
        ibouy.route.iTWD_iBy(iphase)=windInfo.f.iF_iTWD(ibouy.route.xyCoord(i,1),..
```



```
            ibouy.route.xyCoord(i,2),indexTs);
        ibouy.route.iTWS_iBy(iphase)=windInfo.f.iF_iTWS(ibouy.route.xyCoord(i,1),..
            ibouy.route.xyCoord(i,2),indexTs); % [kn]
        ibouy.route.ang2iBy2iTWD(iphase)=(180-ibouy.route.iTWD_iBy(iphase))-...
            (180-ibouy.route.ang2ibuoy(iphase,1));

    elseif compInfo.FieldType==1
        indexTs=compInfo.startCompTime+compInfo.t0min(iphase);
        ibouy.route.iTWD_iBy(iphase)=TWD;
        ibouy.route.iTWS_iBy(iphase)=TWS;
        ibouy.route.ang2iBy2iTWD(iphase)=(180-ibouy.route.iTWD_iBy(iphase))-...
            (180-ibouy.route.ang2ibuoy(iphase,1));

    elseif compInfo.FieldType==21  % 1= constant, 2= matrix_NetCDF, 3= GRID and 4=V
        indexTs=compInfo.startCompTimeIdx+round(compInfo.t0min(iphase)/(compInfo.mi
        if indexTs0(iphase)<indexTs || round(indexTs)==indexTs

            % Estimate the velocity of the wind speed at the end buoy [u v]
            % calcualte the velocities at the start and end -- REVIEW maybe only ne
            ibouy.route.uviTWS_iBy(iphase,:)= [griddata(windInfo.xTWS(:,:,indexTs),
                windInfo.uo_avgTWS_iby(:,:,indexTs),OptData.buoy_start(iphase,1),..
                OptData.buoy_start(iphase,2),'natural'),...
                griddata(windInfo.xTWS(:,:,indexTs),windInfo.yTWS(:,:,indexTs),wind
                OptData.buoy_start(iphase,1),OptData.buoy_start(iphase,2),'natural'
        elseif indexTs0(iphase-1)==indexTs
            ibouy.route.uviTWS_iBy(iphase,:)=ibouy.route.uvTWS_iBy(iphase-1,:);
        else
            itime_index=floor(indexTs);
            ftime_index=ceil(indexTs);

            ibouy.route.uviTWS_iBy(iphase,:)=...
                [interp1([itime_index;ftime_index],[griddata(windInfo.xTWS(:,:,itim
                windInfo.yTWS(:,:,itime_index),windInfo.uo_avgTWS_iby(:,:,itime_ind
                OptData.buoy_start(iphase,1),OptData.buoy_start(iphase,2),'natural'
                griddata(windInfo.xTWS(:,:,ftime_index),windInfo.yTWS(:,:,ftime_ind
                windInfo.uo_avgTWS_iby(:,:,ftime_index),OptData.buoy_start(iphase,1
                OptData.buoy_start(iphase,2),'natural')],indexTs),...
                interp1([itime_index;ftime_index],[griddata(windInfo.xTWS(:,:,itime
                windInfo.yTWS(:,:,itime_index), windInfo.vo_avgTWS_iby(:,:,itime_in
                OptData.buoy_start(iphase,1),OptData.buoy_start(iphase,2),'natural'
                griddata(windInfo.xTWS(:,:,ftime_index),windInfo.yTWS(:,:,ftime_ind
                windInfo.vo_avgTWS_iby(:,:,ftime_index),OptData.buoy_start(iphase,1
                OptData.buoy_start(iphase,2),'natural')],indexTs)];
        end

        ibouy.route.iTWD_iBy(iphase)=-180+angNW(atan2d(ibouy.route.uviTWS_iBy(iphas
            ibouy.route.uviTWS_iBy(iphase,1)));
        ibouy.route.iTWS_iBy(iphase)=vecnorm(ibouy.route.uviTWS_iBy(iphase,:),2,2)/
        % Angle of the buoy cw and CCW if the result angle is negative it means is
        ibouy.route.ang2iBy2iTWD(iphase)=(180-ibouy.route.iTWD_iBy(iphase))-...
            (180-ibouy.route.ang2ibuoy(iphase,1));


    end
    if ibouy.route.ang2iBy2iTWD(iphase)==360 || ibouy.route.ang2iBy2iTWD(iphase)==-
        ibouy.route.ang2iBy2iTWD(iphase)=0;
    elseif ibouy.route.ang2iBy2iTWD(iphase)>360
        ibouy.route.ang2iBy2iTWD(iphase)=rem(ibouy.route.ang2iBy2iTWD(iphase),360);
    elseif ibouy.route.ang2iBy2iTWD<-360
        ibouy.route.ang2iBy2iTWD(iphase)=rem(ibouy.route.ang2iBy2iTWD(iphase),-360)
    end

    OptData.angle_wind_mark(iphase)=sqrt((ibouy.route.ang2iBy2iTWD(iphase)).^2);
    OptData.wind_mark(iphase) = OptData.angle_wind_mark(iphase);

    %  % Detect upwind leg for right initial guess:
    if OptData.wind_mark(iphase)<=45
        OptData.wind_mark_cond(iphase)="upw";
        % Determine interpolation or extrapolation approach, these depends
        % on the wind speed and the velocity.

        anglesr =[0:0.2:180, OptData.wind_mark(iphase)];
        max_crit=cosd(anglesr);

    elseif OptData.wind_mark(iphase) >45 && OptData.wind_mark(iphase)<=135
        OptData.wind_mark_cond(iphase)="reach";

        anglesr=[45:0.2:135, OptData.wind_mark(iphase)];
        max_crit= sind(anglesr);
    else
        OptData.wind_mark_cond(iphase)="down";
        % anglesr = 135:0.2:180;
        anglesr=[135:0.2:180, OptData.wind_mark(iphase)];
        max_crit=cosd(anglesr);
    end
    % VMG Direction and Velocity
    if ibouy.route.iTWS_iBy(iphase) < min(velInfo.vpp_wind_speed)
        psi_port0 = anglesr(max_crit.*interp2(velInfo.vpp_wind_speed,velInfo.vpp_wi
            velInfo.vpp_table,bsxfun(@times,ibouy.route.iTWS_iBy(iphase),ones(size(
            ==max(max_crit.*interp2(velInfo.vpp_wind_speed,velInfo.vpp_wind_angle,.
            velInfo.vpp_table,bsxfun(@times,ibouy.route.iTWS_iBy(iphase),ones(size(

        OptData.u_port0(iphase) = interp2(velInfo.vpp_wind_speed,velInfo.vpp_wind_a
            velInfo.vpp_table,ibouy.route.iTWS_iBy(iphase),psi_port0,'makima');
    else
        psi_port0 = anglesr(max_crit.*interp2(velInfo.vpp_wind_speed,...
            velInfo.vpp_wind_angle,velInfo.vpp_table,bsxfun(@times,...
            ibouy.route.iTWS_iBy(iphase),ones(size(anglesr))),...
            anglesr)==max(max_crit.*interp2(velInfo.vpp_wind_speed,velInfo.vpp_wind
            velInfo.vpp_table,bsxfun(@times,ibouy.route.iTWS_iBy(iphase),ones(size(
        OptData.u_port0(iphase) = interp2(velInfo.vpp_wind_speed,velInfo.vpp_wind_a
            velInfo.vpp_table,ibouy.route.iTWS_iBy(iphase),psi_port0);
    end
    OptData.u_port_projected0(iphase) = cosd(psi_port0)*OptData.u_port0(iphase); %
    %     psi_starboard0 = psi_port0;
    %     psi_port0=-psi_port0;
    %
```

```
    %     OptData.psi0_1(iphase)= -psi_port0;  %port left hand from the boat
    %     OptData.psif_1(iphase)= psi_port0;  %startboard right hand from the boat
    %     %  1= LeftHand/Port Attack; 2=RightHand/Startboar Attack
    OptData.psi(1,iphase)=-psi_port0; % Port
    OptData.psi(2,iphase)= psi_port0; % Startboardor
    OptData.next_mark(iphase) = psi_port0;
    %         next_mark(iphase,:) = rhumb_angle_max;
    %         psi0_1(iphase,1) = 180-rhumb_angle_max;
    %         psif_1(iphase,1) = rhumb_angle_max; %psi0_1(iphase,1);
```

Angle Direction for the next buoy

```
    OptData.angle_port(1,iphase)=ibouy.route.iTWD_iBy(iphase)+OptData.psi(1,iphase)
    OptData.angle_starb(1,iphase)=ibouy.route.iTWD_iBy(iphase)+OptData.psi(2,iphase
    % Port
    if OptData.angle_port(1,iphase)<0
        OptData.angle_port(2,iphase)=360+OptData.angle_port(1,iphase);
    elseif OptData.angle_port(iphase)>360
        OptData.angle_port(2,iphase)=rem(OptData.angle_port(1,iphase),360);
    else
        OptData.angle_port(2,iphase)=OptData.angle_port(1,iphase);
    end
    % Startboard
    if  OptData.angle_starb(1,iphase)<0
        OptData.angle_starb(2,iphase)=360+ OptData.angle_starb(1,iphase);
    elseif  OptData.angle_starb(1,iphase)>360
        OptData.angle_starb(2,iphase)=rem(OptData.angle_starb(1,iphase),360);
    else
        OptData.angle_starb(2,iphase)= OptData.angle_starb(1,iphase);
    end

    %OptParam.attack=2; %  1= LeftHand/Port Attack; 2=RightHand/Startboar Attack
    OptData.attack(1).Dir(:,iphase)=OptData.angle_port(:,iphase);
    OptData.attack(2).Dir(:,iphase)=OptData.angle_starb(:,iphase);
```

Generate a default set of a way points initial condition-for port

```
    % Straight Line
```

```
    ppts(1).xWayPoints(:,iphase)=ones(OptParam.numWayPoints+2,1)*OptData.buoy_start
        linspace(0,OptData.sizeX(iphase),OptParam.numWayPoints+2)';
    ppts(1).yWayPoints(:,iphase)=ones(OptParam.numWayPoints+2,1)*OptData.buoy_start
        linspace(0,OptData.sizeY(iphase),OptParam.numWayPoints+2)';

    % OptParam.attack=2; % 1= LeftHand/Port Attack; 2=RightHand/Startboar Attack
    if OptData.wind_mark_cond(iphase)=="upw"
        % % VMG approach points and Path generation

        alphaprim=ibouy.route.ang2ibuoy(iphase,1)-OptData.attack(OptParam.attack).D

        ytemp=tand(alphaprim)*OptData.sizeY(iphase);%+OptData.sizeY(iphase);
        ytemp2=OptData.buoy_start(iphase,2)-ytemp;
        ppts(2).xWayPoints(1:ceil(OptParam.numWayPoints/2)+1,iphase)=...
            ones(ceil(OptParam.numWayPoints/2)+1,1)*...
            OptData.buoy_start(iphase,1)+...
            linspace(0,OptData.sizeX(iphase),ceil(OptParam.numWayPoints/2)+1)';
        ppts(2).xWayPoints(ceil(OptParam.numWayPoints/2)+2:OptParam.numWayPoints+2,
            OptData.buoy_end(iphase,1);

        ppts(2).yWayPoints(1:ceil(OptParam.numWayPoints/2)+1,iphase)=...
            ones(ceil(OptParam.numWayPoints/2)+1,1)*...
            OptData.buoy_start(iphase,2)-...
            linspace(0,ytemp-OptData.sizeY(iphase),ceil(OptParam.numWayPoints/2)+1)

        ppts(2).yWayPoints(ceil(OptParam.numWayPoints/2)+2:OptParam.numWayPoints+2,
            ones(ceil(OptParam.numWayPoints/2),1)*(OptData.buoy_start(iphase,2)+Opt
            linspace(0,ytemp,ceil(OptParam.numWayPoints/2))';


        betaprima=90-alphaprim;
        xprima=sind(betaprima)*ibouy.route.Dist2iBy(iphase,1);
        xprima_proj=xprima*cosd(alphaprim);
        yprima_proj=xprima*sind(alphaprim);


        ppts(3).xWayPoints(1:ceil(OptParam.numWayPoints/2)+1,iphase)=...
            ones(ceil(OptParam.numWayPoints/2)+1,1)*...
            OptData.buoy_start(iphase,1)+...
            linspace(0,xprima_proj,ceil(OptParam.numWayPoints/2)+1)';

        ppts(3).yWayPoints(1:ceil(OptParam.numWayPoints/2)+1,iphase)=...
            ones(ceil(OptParam.numWayPoints/2)+1,1)*...
            OptData.buoy_start(iphase,2)+...
            linspace(0,yprima_proj,ceil(OptParam.numWayPoints/2)+1)';

        ppts(3).xWayPoints(ceil(OptParam.numWayPoints/2)+2:OptParam.numWayPoints+2,
            ones(ceil(OptParam.numWayPoints/2),1)*...
            (OptData.buoy_start(iphase,1)+xprima_proj)+...
            linspace(0,OptData.buoy_end(iphase,1)-(OptData.buoy_start(iphase,1)+xpr
            ceil(OptParam.numWayPoints/2))';

        ppts(3).yWayPoints(ceil(OptParam.numWayPoints/2)+2:OptParam.numWayPoints+2,
            ones(ceil(OptParam.numWayPoints/2),1)*...
            ppts(3).yWayPoints(ceil(OptParam.numWayPoints/2)+1,iphase)+...
            linspace(0,OptData.buoy_end(iphase,2)-ppts(3).yWayPoints(ceil(OptParam.
            ceil(OptParam.numWayPoints/2))';

    else
        %  % VMG approach points and Path generation MidPoint turn
        ppts(2).xWayPoints(1:OptParam.numWayPoints+2,iphase)=ones(OptParam.numWayPo
            OptData.buoy_start(iphase,1)+linspace(0,OptData.sizeX(iphase),OptParam.

        ppts(2).yWayPoints(1:ceil(OptParam.numWayPoints/2)+1,iphase)=...
            ones(ceil(OptParam.numWayPoints/2)+1,1)*OptData.buoy_start(iphase,2)+..
            linspace(0,OptData.sizeY(iphase),ceil(OptParam.numWayPoints/2)+1)';
```

```
        ppts(2).yWayPoints(ceil(OptParam.numWayPoints/2)+2:OptParam.numWayPoints+2,
            ones(ceil(OptParam.numWayPoints/2),1)*...
            ppts(2).yWayPoints(ceil(OptParam.numWayPoints/2)+1,iphase)+...
            linspace(0,(OptData.buoy_end(iphase,2)-ppts(2).yWayPoints(ceil(OptParam
            ceil(OptParam.numWayPoints/2))'; %OptParam.numWayPoints+2-(ceil(OptPara


        ppts(3).xWayPoints(1:ceil(OptParam.numWayPoints/2),iphase)=ones(ceil(OptPar
            OptData.buoy_start(iphase,1)+linspace(0,OptData.sizeY(iphase)*...
            sind(OptData.attack(OptParam.attack).Dir(2,iphase)),ceil(OptParam.numWa

        ppts(3).xWayPoints(ceil(OptParam.numWayPoints/2)+1:OptParam.numWayPoints+2,
            ones(ceil(OptParam.numWayPoints/2)+1,1)*...
            ppts(3).xWayPoints(ceil(OptParam.numWayPoints/2),iphase)+...
            linspace(0,(OptData.buoy_end(iphase,1)-ppts(3).xWayPoints(ceil(OptParam
            ceil(OptParam.numWayPoints/2)+1)';

        ppts(3).yWayPoints(1:ceil(OptParam.numWayPoints/2),iphase)=ones(ceil(OptPar
            OptData.buoy_start(iphase,2)+...
            linspace(0,(0.5*ibouy.route.Dist2iBy(iphase,1)*cosd(ibouy.route.ang2ibu
            ceil(OptParam.numWayPoints/2))';

        ppts(3).yWayPoints(ceil(OptParam.numWayPoints/2)+1:OptParam.numWayPoints+2,
            ones(ceil(OptParam.numWayPoints/2)+1,1)*...
            ppts(3).yWayPoints(ceil(OptParam.numWayPoints/2),iphase)+...
            linspace(0,(OptData.buoy_end(iphase,2)-ppts(3).yWayPoints(ceil(OptParam
            iphase)),ceil(OptParam.numWayPoints/2)+1)';

    end
```

Bound per Leg

```
    for i=1:3
        PointsAlt(iphase).minWayPointsLeg(i,1)=min(ppts(i).xWayPoints(:,iphase));
        PointsAlt(iphase).minWayPointsLeg(i,2)=min(ppts(i).yWayPoints(:,iphase));
        PointsAlt(iphase).maxWayPointsLeg(i,1)=max(ppts(i).xWayPoints(:,iphase));
        PointsAlt(iphase).maxWayPointsLeg(i,2)=max(ppts(i).yWayPoints(:,iphase));
        PointsAlt(iphase).meanWayPointLeg(i,1)= mean(ppts(i).xWayPoints(:,iphase));
        PointsAlt(iphase).meanWayPointLeg(i,2)= mean(ppts(i).yWayPoints(:,iphase));
    end

    PointsAlt(iphase).ctrWayPoint=[mean(PointsAlt(iphase).meanWayPointLeg(:,1)),...
        mean(PointsAlt(iphase).meanWayPointLeg(:,2))];

    PointsAlt(iphase).lbWayPoint=PointsAlt(iphase).ctrWayPoint -...
        [PointsAlt(iphase).ctrWayPoint(:,1)-...
        min([PointsAlt(iphase).minWayPointsLeg(:,1);PointsAlt(iphase).maxWayPointsL
        PointsAlt(iphase).ctrWayPoint(:,2)-min([PointsAlt(iphase).minWayPointsLeg(:
        PointsAlt(iphase).maxWayPointsLeg(:,2)])].*[OptData.tolx, OptData.toly]-[10

    PointsAlt(iphase).ubWayPoint=[max([PointsAlt(iphase).minWayPointsLeg(:,1);...
        PointsAlt(iphase).maxWayPointsLeg(:,1)])-PointsAlt(iphase).ctrWayPoint(:,1)
        max([PointsAlt(iphase).minWayPointsLeg(:,2);PointsAlt(iphase).maxWayPointsL
        PointsAlt(iphase).ctrWayPoint(:,2)].*[OptData.tolx, OptData.toly] + PointsA

    PointsAlt(iphase).lb = reshape([ PointsAlt(iphase).lbWayPoint(:,1)*ones(1,OptPa
        PointsAlt(iphase).lbWayPoint(:,2)*ones(1,OptParam.numWayPoints)],[],1);
    PointsAlt(iphase).ub = reshape([PointsAlt(iphase).ubWayPoint(:,1)*ones(1,OptPar
        PointsAlt(iphase).ubWayPoint(:,2)*ones(1,OptParam.numWayPoints)],[],1);

    PointsAlt(iphase).ub_v=reshape(PointsAlt(iphase).ub,2,[])';
    PointsAlt(iphase).lb_v=reshape(PointsAlt(iphase).lb,2,[])';
```

Temo Graph by PReliminary Paths

```
    markTemp={'-+r',':ob','-.*g'};
    figure
    for i=1:3
        plot (ppts(i).xWayPoints(:,iphase), ppts(i).yWayPoints(:,iphase),markTemp{i
        hold on
        grid on
        grid minor
    end

    hold on
    plot(PointsAlt(iphase).ctrWayPoint(:,1), PointsAlt(iphase).ctrWayPoint(:,2),'*r

    hold on
    plot(PointsAlt(iphase).lbWayPoint(:,1),PointsAlt(iphase).lbWayPoint(:,2),'+m')

    hold on
    plot(PointsAlt(iphase).ubWayPoint(:,1),PointsAlt(iphase).ubWayPoint(:,2),'+m')
    %Review the limits
    hold on
    plot([PointsAlt(iphase).ub_v(:,1);PointsAlt(iphase).lb_v(:,1)],...
        [PointsAlt(iphase).ub_v(:,2);PointsAlt(iphase).lb_v(:,2)],':ob')
```

Link node w/time

```
    if iphase>1
        InfoLink(iphase,:).xy_coord=Infolink(iphase-1,:).dlink.time(1,1:2);
        InfoLink(iphase,:).time=Infolink(iphase-1,:).dlink.time(1,3);
        % InfoLink(iphase,:).iTWS=Infolink(iphase-1,:).dlink.iTWS;
        InfoLink(iphase,:).TWS=Infolink(iphase-1,:).dlink.TWS;
        InfoLink(iphase,:).TWD=Infolink(iphase-1,:).dlink.TWD;
        InfoLink(iphase,:).TWD_iby=Infolink(iphase-1,:).dlink.TWD_iby;
        buoy_link(iphase,:)=[InfoLink(iphase-1,:).xy_coord];%xy_ibuoyRoute(iphase-1
        link_cond(iphase)=OptData.wind_mark_cond(iphase-1);
    else
    end
```

Estimate the time the preliminar set of points

Estimate the time following a straight Line (SL) Generate a continuos path from the waypoints for a Straight line

```
    for i=1:3
```

```
        WayPoints(iphase,i).PathPoints = WayPoints_To_Path([ppts(i).xWayPoints(:,ip
            ppts(i).yWayPoints(:,iphase)],'linear',OptData.buoy_start(iphase,:),...
            OptData.buoy_end(iphase,:),OptParam.interval+1);

        WayPoints(iphase,i).TravelTime = getTimeFromPath_iMP(WayPoints(iphase,i).Pa
            compInfo,windInfo,OptData.buoy_start(iphase,:),OptData.buoy_end(iphase,
            CONSTANTS,velInfo,OptParam.interval,iphase);
    end
```

Get the Times for the preliminary Paths

```
    for i=1:3
        PointsAlt(iphase).time(i)=[WayPoints(iphase,i).TravelTime];
        PointsAlt(iphase).x(:,i)=ppts(i).xWayPoints(:,iphase);
        PointsAlt(iphase).y(:,i)=ppts(i).yWayPoints(:,iphase);

    end

    [~, PointsAlt(iphase).minIdx]=mink(PointsAlt(iphase).time,2);
```

Struct data points

```
    for i=1:3
        PointsAlt(iphase).time(i)=[WayPoints(iphase,i).TravelTime];
        PointsAlt(iphase).x(:,i)=ppts(i).xWayPoints(:,iphase);
        PointsAlt(iphase).y(:,i)=ppts(i).yWayPoints(:,iphase);
    end
    [~, PointsAlt(iphase).minIdx]=mink(PointsAlt(iphase).time,2);
```

Define the Objective Function

```
    objectiveFun=@(P) getTimeFromPath_iMP(P,compInfo,windInfo,OptData.buoy_start(ip
        OptData.buoy_end(iphase,:),'linear',CONSTANTS,velInfo,OptParam.interval,iph

    % Define the function for the constraints
    constraintsFun=@(P) constraints_iMP(P,compInfo,windInfo,OptData.buoy_start(ipha
        OptData.buoy_end(iphase,:),buoy_link(iphase,:),OptData.wind_mark_cond(iphas
        string(link_cond(iphase)),velInfo,CONSTANTS,InfoLink(iphase,:),iphase);
```

DEfine the constraints from the 2 min time values

```
    PointsAlt(iphase).ic=NaN((OptParam.numWayPoints*2),3,'double');
    PointsAlt(iphase).optTime=NaN(1,3,'double');
    PointsAlt(iphase).optCompEff=NaN(1,3,'double');
    PointsAlt(iphase).xoptWayPoints=NaN((OptParam.numWayPoints+2),3,'double');
    PointsAlt(iphase).yoptWayPoints=NaN((OptParam.numWayPoints+2),3,'double');
    %  PointsAlt(iphase).optXppts=NaN(1,3,'double');
```

```
    for i=1:2
        ic_temp=[ppts(PointsAlt(iphase).minIdx(i)).xWayPoints(2:end-1,iphase)';...
            ppts(PointsAlt(iphase).minIdx(i)).yWayPoints(2:end-1,iphase)'];
```

```
else
    ibY="LeftPnt";
end

if OptParam.attack==1
    attck="LeftHnd";
else
    attck="RightHnd";
end

if compInfo.FieldType==2
    windF="NetCDFWind";
elseif compInfo.FieldType==3
    windF="GridWind";
elseif compInfo.FieldType==4
    windF="SapSailMeas";
elseif compInfo.FieldType==21
    windF=strcat("NetCDFWind_m",num2str(compInfo.constTimeStep));
else
    windF="ConstWind";
end

if compInfo.current==0
    currentF="NoCurrent";
else
    currentF="WithCurrent";
end

startTimestr=strcat(num2str(compInfo.startCompTimeHR),':',num2str(compInfo.startCom
DateFileCreated=datestr(datetime,'dd-mmmyy-HHMMSS');
nameOptimResFile=strcat('Results-',DateFileCreated,'-MaxFunEval-',num2str(opts.MaxF
    '-MaxIter-',num2str(opts.MaxIter),'-',ibY,'-',attck,'-',currentF,'-',windF,'_',

SaveFolder=strcat(file.mydir,'/Results_report/results/',datestr(datetime,'dd-mmmyy'
if not(exist(SaveFolder,'dir'))
    mkdir(SaveFolder)
end
```

Save the results on mat file%

```
filename_loc=strcat(SaveFolder,'/',nameOptimResFile);
save(filename_loc,'ibouy','compInfo','windInfo','leg_mtp','InfoLink','Infolink',...
    'opts','PointsAlt', 'velInfo','file','OptParam','OptData')
```

PLOTS

Prelocation

```
col={'--','+-','+:','*:','o-.','x-.'};
p=zeros();
l={};

figure ('name','MinWayPoints(iphase).TimePath')
```

```
plot(ibouy.route.xyCoord(:,1),ibouy.route.xyCoord(:,2),'xr','markersize',10);
for i=1:size(leg_mtp,2)
    hold all
    p(i)=plot(leg_mtp(i).x,leg_mtp(i).y,'linewidth',1.7);

    l(i,1)=strcat('Leg',{' '},num2str(i),':',{' '},num2str(leg_mtp(i).sec2min),{' '
        num2str(leg_mtp(i).secdec),{' '},'sec');
    %     plot([PointsAlt(i).ub_v(:,1);PointsAlt(i).lb_v(:,1)],[PointsAlt(i).ub_v(:
end
grid on
grid minor
tl=title(strcat('Optimal Path, starting at'," ",startTimestr," ",num2str(OptParam.n
xl=xlabel('Longitud in meters');
yl=ylabel('Latitud in meters');
lg=legend(p,l,'Location','best','FontSize',12);
text(ibouy.route.xyCoord(:,1),ibouy.route.xyCoord(:,2),strsplit(num2str(1:noiBy))',
    'HorizontalAlignment','center','FontWeight','Bold','FontSize',10);

% Review the limits
hold on
% OptParam.ub_v=reshape (OptParam.ub,2,[])';
% OptParam.lb_v=reshape(OptParam.lb,2,[])';


% annotation-Position
ax1 = axes('Position',[0 0 1 1],'Visible','off');
ax2 = axes('Position',[.3 .1 .6 .8],'Visible','off');
axes(ax1) % sets ax1 to current axes
txt=text(.002,0.002,nameOptimResFile);

% dim = [.25 .001 .2 .2];

% % % % Latex title
set(tl,'FontSize',22,'Interpreter','latex');
set(xl,'FontSize',18,'Interpreter','latex');
set(yl,'FontSize',18,'Interpreter','latex');
set(lg,'FontSize',14,'Interpreter','latex');
set(txt,'FontSize',9,'Interpreter','latex');

% Enlarge figure to full screen.

%set(gcf, 'Units', 'Normalized', 'OuterPosition', [0, 0.04, 1, 0.96]);
set(gcf,'PaperUnits','inches')%'PaperPosition',[0 0 4 3])
print('-bestfit','BestFitFigure','-dpdf')

% % % % Svave pictures
namefig=strcat('MinWayPoints(iphase)_rght.TimePath2',nameOptimResFile,'.pdf');
figsave=strcat(SaveFolder,'/',namefig);
saveas(gcf,figsave)
```

Plots 4 alternatives

```
slp=zeros();
```

```
else
    ibY="LeftPnt";
end

if OptParam.attack==1
    attck="LeftHnd";
else
    attck="RightHnd";
end

if compInfo.FieldType==2
    windF="NetCDFWind";
elseif compInfo.FieldType==3
    windF="GridWind";
elseif compInfo.FieldType==4
    windF="SapSailMeas";
elseif compInfo.FieldType==21
    windF=strcat("NetCDFWind_m",num2str(compInfo.constTimeStep));
else
    windF="ConstWind";
end

if compInfo.current==0
    currentF="NoCurrent";
else
    currentF="WithCurrent";
end

startTimestr=strcat(num2str(compInfo.startCompTimeHR),':',num2str(compInfo.startCom
DateFileCreated=datestr(datetime,'dd-mmmyy-HHMMSS');
nameOptimResFile=strcat('Results-',DateFileCreated,'-MaxFunEval-',num2str(opts.MaxF
    '-MaxIter-',num2str(opts.MaxIter),'-',ibY,'-',attck,'-',currentF,'-',windF,'_',

SaveFolder=strcat(file.mydir,'/Results_report/results/',datestr(datetime,'dd-mmmyy'
if not(exist(SaveFolder,'dir'))
    mkdir(SaveFolder)
end
```

Save the results on mat file%

```
filename_loc=strcat(SaveFolder,'/',nameOptimResFile);
save(filename_loc,'ibouy','compInfo','windInfo','leg_mtp','InfoLink','Infolink',...
    'opts','PointsAlt', 'velInfo','file','OptParam','OptData')
```

PLOTS

Prelocation

```
col={'--','+-','+:','*:','o-.','x-.'};
p=zeros();
l={};

figure ('name','MinWayPoints(iphase).TimePath')
plot(ibouy.route.xyCoord(:,1),ibouy.route.xyCoord(:,2),'xr','markersize',10);
for i=1:size(leg_mtp,2)
    hold all
    p(i)=plot(leg_mtp(i).x,leg_mtp(i).y,'linewidth',1.7);

    l(i,1)=strcat('Leg',{' '},num2str(i),':',{' '},num2str(leg_mtp(i).sec2min),{' '
        num2str(leg_mtp(i).secdec),{' '},'sec');
    %     plot([PointsAlt(i).ub_v(:,1);PointsAlt(i).lb_v(:,1)],[PointsAlt(i).ub_v(:
end
grid on
grid minor
tl=title(strcat('Optimal Path, starting at'," ",startTimestr," ",num2str(OptParam.n
xl=xlabel('Longitud in meters');
yl=ylabel('Latitud in meters');
lg=legend(p,l,'Location','best','FontSize',12);
text(ibouy.route.xyCoord(:,1),ibouy.route.xyCoord(:,2),strsplit(num2str(1:noiBy))',
    'HorizontalAlignment','center','FontWeight','Bold','FontSize',10);

% Review the limits
hold on
% OptParam.ub_v=reshape (OptParam.ub,2,[])';
% OptParam.lb_v=reshape(OptParam.lb,2,[])';


% annotation-Position
ax1 = axes('Position',[0 0 1 1],'Visible','off');
ax2 = axes('Position',[.3 .1 .6 .8],'Visible','off');
axes(ax1) % sets ax1 to current axes
txt=text(.002,0.002,nameOptimResFile);

% dim = [.25 .001 .2 .2];

% % % % Latex title
set(tl,'FontSize',22,'Interpreter','latex');
set(xl,'FontSize',18,'Interpreter','latex');
set(yl,'FontSize',18,'Interpreter','latex');
set(lg,'FontSize',14,'Interpreter','latex');
set(txt,'FontSize',9,'Interpreter','latex');

% Enlarge figure to full screen.

%set(gcf, 'Units', 'Normalized', 'OuterPosition', [0, 0.04, 1, 0.96]);
set(gcf,'PaperUnits','inches')%'PaperPosition',[0 0 4 3])
print('-bestfit','BestFitFigure','-dpdf')

% % % % Svave pictures
namefig=strcat('MinWayPoints(iphase)_rght.TimePath2',nameOptimResFile,'.pdf');
figsave=strcat(SaveFolder,'/',namefig);
saveas(gcf,figsave)
```

Plots 4 alternatives

```
slp=zeros();
```

```
wp=zeros();
wp2=zeros();
optsl=zeros();
optvmg=zeros();
optvmg2=zeros();
slt={};
wt={};
wt2={};
opslt={};
opvmgt={};
opvmgt2={};
```

% % % % Plots buoys location mid points travel

```
figure ('name','WayPoints(iphase).TimePath-alt')
plot(ibouy.route.xyCoord(:,1),ibouy.route.xyCoord(:,2),'sr','markersize',15);
text(ibouy.route.xyCoord(:,1),ibouy.route.xyCoord(:,2),strsplit(num2str(1:noiBy))',
    'HorizontalAlignment','center','FontWeight','Bold','FontSize',10)
for j=1:noiBy % Alternative
    for i=1:noiBy-1 % leg
        hold all
        slp(i,j)=plot(PointsAlt(i).xCoordleg(:,j),PointsAlt(i).yCoordleg(:,j),col{j
        slt(i,j)=strcat('Leg-',num2str(i),{' '},legID(j),':',{' '},num2str(PointsAl
        %         plot([PointsAlt(i).ub_v(:,1);PointsAlt(i).lb_v(:,1)],[PointsAlt(i
    end
end
grid on
grid minor
tl=title(strcat('Paths Developed, starting at'," ",startTimestr),'FontSize',12);
xl=xlabel('longitud in meters');
yl=ylabel('latitud in meters');

apt=reshape(slp,[],1);
alt=reshape(slt,[],1);

lg=legend(apt,alt,'Location','bestoutside');
hold on
if compInfo.FieldType ~=1
    plot(windInfo.x_ll,windInfo.y_ll,'ok')
end
%Review the limits
hold on
%Annotation position and style
ax1 = axes('Position',[0 0 1 1],'Visible','off');
%ax2 = axes('Position',[.3 .1 .6 .8],'Visible','off');
axes(ax1) % sets ax1 to current axes
txt=text(.02,0.02,nameOptimResFile);

% Latex title
set(tl,'FontSize',20,'Interpreter','latex');
set(xl,'FontSize',14,'Interpreter','latex');
set(yl,'FontSize',14,'Interpreter','latex');
set(lg,'FontSize',10,'Interpreter','latex');
```

```
set(txt,'FontSize',7,'Interpreter','latex');

print('-bestfit','BestFitFigure','-dpdf');

% Save pictures
namefig=strcat('WayPoints(iphase).TimePath-alt2',nameOptimResFile,'.pdf');
figsave=strcat(SaveFolder,'/',namefig);
saveas(gcf,figsave)
% %
```

Table -time results

```
timet=struct2table(leg_mtp);
%strcat(strtrim(cellstr(num2str(velInfo.vpp_wind_speed(:)))),'kn');
tt_row=cellstr(strcat('leg_',num2str([1:size(leg_mtp,2)]')));
Total_sec=timet.time;
minutes=timet.sec2min;
secDec=timet.secdec;
tt=table(Total_sec,minutes,secDec,'RowName',tt_row);
tt.Total_sec(end+1,1)=sum(tt.Total_sec);

if compInfo.FieldType==2
    tt.minutes(end,1)=floor(tt.Total_sec(end,1)/60);
    tt.secDec(end,1)=rem(tt.Total_sec(end,1),60);
elseif compInfo.FieldType==4
    tt.minutes(end,1)=floor(tt.Total_sec(end,1)/compInfo.min2sec);
    tt.secDec(end,1)=rem(tt.Total_sec(end,1),compInfo.min2sec);
elseif compInfo.FieldType==21
    tt.minutes(end,1)=floor(tt.Total_sec(end,1)/60);
    tt.secDec(end,1)=rem(tt.Total_sec(end,1),60);
end

tt.Properties.RowNames(6,1)={'RouteTime'};
figure ('name','Min_Times Sort')
uitable('Data',tt{:,:},'ColumnName',tt.Properties.VariableNames,'RowName',tt.Proper
    'Units', 'Normalized', 'Position',[0, 0, 1, 1],'FontSize',16);
%set(cgf,'Name','myName')
txt_title = uicontrol('Style', 'text', 'Position', [10 220 500 30], 'String', strca
%Annotation position and style
ax1 = axes('Position',[0 0 1 1],'Visible','off');
ax2 = axes('Position',[.3 .1 .6 .8],'Visible','off');
axes(ax1) % sets ax1 to current axes
txt=text(.02,0.02,nameOptimResFile);

% print('-bestfit','BestFitFigure','-dpdf');

% Save pictures
namefig=strcat('Min_Times_Sort',nameOptimResFile,'.pdf');
figsave=strcat(SaveFolder,'/',namefig);
saveas(gcf,figsave)
```

Plots by leg 4 alternatives