THE UNIVERSITY
of EDINBURGH

Dynamic yacht strategy optimisation

Francesca Tagliaferri

Doctor of Philosophy

THE UNIVERSITY OF EDINBURGH

2015

To my family,
present, past and future.

Abstract

Yacht races are won by good sailors racing fast boats. A good skipper takes decisions at key moments of the race based on the anticipated wind behaviour and on his position on the racing area and with respect to the competitors. His aim is generally to complete the race before all his opponents, or, when this is not possible, to perform better than some of them. In the past two decades some methods have been proposed to compute optimal strategies for a yacht race. Those strategies are aimed at minimizing the expected time needed to complete the race and are based on the assumption that the faster a yacht, the higher the number of races that it will win (and opponents that it will defeat). In a match race, however, only two yachts are competing. A skipper's aim is therefore to complete the race before his opponent rather than completing the race in the shortest possible time. This means that being on average faster may not necessarily mean winning the majority of races. This thesis sets out to investigate the possibility of computing a sailing strategy for a match race that can defeat an opponent who is following a fixed strategy that minimises the expected time of completion of the race. The proposed method includes two novel aspects in the strategy computation:

- A short-term wind forecast, based on an Artificial Neural Network (ANN) model, is performed in real time during the race using the wind measurements collected on board.
- Depending on the relative position with respect to the opponent, decisions with different levels of risk aversion are computed. The risk attitude is modeled using Coherent Risk Measures.

The proposed algorithm is implemented in a computer program and is tested by simulating match races between identical boats following progressively refined strategies. Results presented in this thesis show how the intuitive idea of taking more risk when losing and having a conservative attitude when winning is confirmed in the risk model used. The performance of ANN for short-term wind forecasting is tested both on wind speed and wind direction. It is shown that for time steps of the order of seconds and adequate computational power ANN perform better than linear models (persistence models, ARMA) and other nonlinear models (Support Vector Machines). The outcome of the simulated races confirms that maximising the probability of winning a match race does not necessarily correspond to minimising the expected time needed to complete the race.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

Francesca Tagliaferri

Contents

Abstract iii

Declaration iv

Figures and Tables viii

Nomenclature xii

1 Introduction 1

1.1 Yacht racing tactics and strategy . . . 3
1.2 Literature review on yacht racing . . . 5
1.3 Problem statement . . . 7
1.4 Thesis outline . . . 8
1.5 Publications . . . 9

I Background 10

2 Background on wind forecasting 11

2.1 Numerical methods . . . 13
2.2 Statistical methods . . . 13
2.2.1 Persistence model . . . 13
2.2.2 Autoregressive Moving Average processes . . . 14
2.3 Artificial Neural Netwoks . . . 15
2.3.1 Literature review on ANN in forecasting . . . 18
2.3.2 Support Vector Machines . . . 20

3 Background on velocity prediction and aerodynamic interaction between two boats 22

3.1 Velocity prediction . . . 22
3.2 Aerodynamic interaction between two boats . . . 25

4 Background on routing algorithms 29

4.1 An introduction to Dynamic Programming . . . 29
4.2 An example of DP execution . . . 30
4.3 Stochastic dynamic programming . . . 32
4.4 Markov Chains . . . 33
4.5 Application of DP to yacht routing . . . 35
4.6 Risk management . . . 36

II Method 40

5 Wind forecasting 41
5.1 Wind dataset . . . 41
5.2 Input and output . . . 45
5.3 ANN model optimisation . . . 45
5.3.1 Optimisation parameters . . . 45
5.3.2 ANN-ensemble forecast . . . 46
5.3.3 Evaluation criteria . . . 46
5.4 Comparison with other models . . . 48
5.4.1 ARMA . . . 48
5.4.2 Support Vector Machines . . . 49
5.5 Markov Chain model . . . 49
5.5.1 Medium-term wind modeling with Markov Chains . . . 49
5.6 Artificial wind time series . . . 50

6 Velocity Prediction and aerodynamic interactions between two boats 52
6.1 Velocity Prediction Program . . . 52
6.1.1 Aerodynamic forces . . . 52
6.1.2 Hydrodynamic forces . . . 55
6.1.3 VPP algorithm . . . 58
6.2 Aerodynamic interactions between two boats . . . 60

7 Routing Algorithms 63
7.1 Grid generation . . . 63
7.2 Tide . . . 66
7.3 Tacking . . . 68
7.4 Risk modeling . . . 70
7.5 Dynamic routing . . . 71

III Results 75

8 Wind forecasting 76
8.1 ANN structure optimisation . . . 76
8.1.1 Wind direction forecasting . . . 76
8.1.2 Wind speed forecasting . . . 79
8.2 Optimum ANN model . . . 82
8.3 Verification of the optimum ANN model . . . 84
8.4 Comparison with other models . . . 87
8.4.1 Persistence model . . . 87
8.4.2 ARMA . . . 88
8.4.3 Support Vector Machines . . . 89

8.5 Summary 90

9 Dynamic routing for a single boat 92

9.1 Constant wind speed and direction 92
9.2 Constant wind speed and increasing wind shift towards one side 93
9.3 Constant wind direction, increasing wind speed 96
9.4 Constant wind speed, temporary wind shift 97
9.5 Recorded wind scenarios 98
9.6 Summary 99

10 Dynamic routing for two boats 101

10.1 Risk management 101
10.2 Example of complete race 106
10.3 Summary 108

IV Concluding Remarks 112

11 Conclusions 113

12 Future work 115

Appendices

A VPP definitions and coefficients 116

Bibliography 118

Figures and Tables

Figures

1.1 Example of polar diagram. 1
1.2 Example of upwind leg. 2
1.3 Example of upwind leg, appeared in Tagliaferri *et al.* (2014). 5

2.1 Schematic representation of a neuron - appeared in Tagliaferri *et al.* (2015). . . . 17
2.2 Example of multi-layer perceptron - appeared in Tagliaferri *et al.* (2015). 17
2.3 Mapping of 2D non linearly separable data to a 3D space where the problem becomes linearly separable - appeared in Tagliaferri *et al.* (2015). 20

3.1 Forces acting on a sailing yacht. 23
3.2 Momentum equilibrium 24
3.3 Schematic representation of flow and pressures around the sails (Viola and Flay, 2011). 25
3.4 Interference zones as described by Johnson (1995). 26
3.5 Bent air effect from Eq. 3.2 27
3.6 Re-elaboration of Figure 3.4 presented in Richards *et al.* (2013) 28

4.1 Example of shortest-path problem. 31
4.2 Example of shortest-path problem. 32
4.3 Representation of a Markov Chain with three states 35
4.4 Example of risk-seeking attitude. 38

5.1 Racing area for the 2013 America's Cup events. 42
5.2 Example of wind direction dataset 44
5.3 Example of wind speed dataset 44
5.4 Representation of the transition matrix for the wind model. 50

6.1 Definitions of sails parameters. 54
6.2 Definition of hull parameters. 55
6.3 Righting moment. 57
6.4 Structure of the loop-based VPP 59
6.5 Comparison between VPP implementation: computation time 60
6.6 Comparison between VPP implementation: results 61
6.7 Wind tunnel test for multiple yachts interactions. Figure from Aubin (2013) 61
6.8 Boat speed variations with a fixed interfering yacht. Figure from Aubin (2013) . . 62

7.1 Racing area 63

7.2 Comparison between grid with fixed spatial steps (left) and with wind-dependent steps (right). 64
7.3 Construction of grid points. 65
7.4 Example of grid construction. 66
7.5 Example of grid construction. 67
7.6 Example of tidal current data available to sailors 67
7.7 Wind triangle taking current into account. 68
7.8 Examples of tack trajectories. 69
7.9 Modified transition matrices for a risk-seeking skipper. Advantageous wind shifts occur with higher probability than disadvantageous ones. (Left) Yacht on the left-hand side of competitor and (Right) yacht on the right-hand side of competitor. . 71
7.10 Transformation matrix for risk-seeking attitude 72
7.11 When boats meet on opposite tacks. 73
7.12 Summary of methodology used. 74

8.1 MAE versus length of the moving average for wind direction (ANN with 20 hidden neurons and 9 minutes input vector). Figure from Tagliaferri *et al.* (2015). 77
8.2 MEI versus length of the moving average for wind direction (ANN with 20 hidden neurons and 9 minutes input vector). Figure from Tagliaferri *et al.* (2015). 77
8.3 MAE versus length of the input vector for wind direction (ANN with 20 hidden neurons and 6 minutes moving average). Figure from Tagliaferri *et al.* (2015). . . 78
8.4 MEI versus length of the input vector for wind direction (ANN with 20 hidden neurons and 6 minutes moving average). Figure from Tagliaferri *et al.* (2015). . . 78
8.5 MAE versus number of neurons for wind direction (ANN 9 minutes input vector and 6 minutes moving average). Figure from Tagliaferri *et al.* (2015). 79
8.6 MEI versus number of neurons for wind direction (ANN 9 minutes input vector and 6 minutes moving average). Figure from Tagliaferri *et al.* (2015). 79
8.7 MAE versus length of the moving average (ANN with 20 hidden neurons and 9 minutes input vector). 80
8.8 MEI versus length of the moving average (ANN with 20 hidden neurons and 9 minutes input vector). 81
8.9 MAE versus length of the input vector (ANN with 20 hidden neurons and 4 minutes moving average). 81
8.10 MEI versus length of the input vector (ANN with 20 hidden neurons and 4 minutes moving average). 82
8.11 MAE versus number of neurons for wind speed (ANN 9 minutes input vector and 4 minutes moving average). 82
8.12 MEI versus number of neurons for wind speed (ANN 9 minutes input vector and 4 minutes moving average). 83
8.13 Example of forecast for the optimised ANN. Red stripes highlight wrong tactical decision, green stripes correct ones. 84

8.14 Example of artificial time series 85
8.15 Example of AR forecast on artificial time series 86
8.16 Example of ANN forecast on artificial time series 86
8.17 Example of AR forecast on artificial time series with suboptimal number of input 87
8.18 Example of AR forecast on artificial time series with suboptimal number of input 87
8.19 MAE versus length of the moving average (SVM). 89
8.20 MEI versus length of the moving average (SVM). 90
8.21 MAE versus length of the input vector (SVM). 90
8.22 MEI versus length of the input vector (SVM). 91

9.1 Grid generated for constant wind direction and constant wind speed 93
9.2 Grid generated for consistent wind shift towards the right. 94
9.3 Grid generated for consistent wind shift towards the left. 95
9.4 Grid generated for increasing wind speed. 96
9.5 Grid generated for consistent wind shift towards the left. 97
9.6 Comparison between strategies computed with different wind knowledge - race 2. 98
9.7 Comparison between strategies computed with different wind knowledge - race 4. 99

10.1 Modified transition matrices for a risk-seeking skipper. Advantageous wind shifts occur with higher probability than disadvantageous ones. (Left) Yacht on the left-hand side of competitor and (Right) yacht on the right-hand side of competitor. . 102
10.2 Races outcomes for risk-neutral attitude. 102
10.3 Races outcomes for optimistic attitude. 103
10.4 Races outcomes for pessimistic attitude. 103
10.5 Time differences for risk-neutral strategy vs optimistic-pessimistic combination. . 104
10.6 Optimal postprocessing matrix 105
10.7 Time differences for risk-neutral strategy vs optimal optimistic-pessimistic combination. 105
10.8 Race example - 1 106
10.9 Race example - 2 107
10.10 Race example - 3 108
10.11 Race example - 4 109
10.12 Race example - 5 110
10.13 Race example - if boats don't see each other 111

Tables

2.1 Forecasting time horizons in different applications 12

4.1 Example of arrival times 37

5.1 Wind data statistics 43
5.2 Correlation coefficients for wind speed and direction 45

6.1 VPP input definition 53

8.1 Performance of optimised ANN model. 83
8.2 Indices for persistence model applied to wind speed 88
8.3 Indices for persistence model applied to wind direction 88
8.4 Comparison between different models for wind direction 88
8.5 Comparison between different model for wind speed 88

9.1 Simulations for increasing wind shifts assuming perfect knowledge of the wind. . 93
9.2 Simulated races with San Francisco wind dataset. 98

A.1 Coefficients for residuary resistance of bare hull, from Keuning and Sonnenberg (1998). 116
A.2 Coefficients for residuary resistance of appendages, from Keuning and Sonnenberg (1998). 116
A.3 Delta resistance hull due to 20°, from Keuning and Sonnenberg (1998). 117
A.4 Coefficients for effective span, from Keuning and Sonnenberg (1998). 117
A.5 Coefficients for aerodynamic lift, from Larsson *et al.* (2014). 117
A.6 Coefficients for aerodynamic drag, from Larsson *et al.* (2014). 117

Nomenclature

A_N	Nominal sail area
A_w	Waterplan area
C	Cost function for SVR, Eq. 2.13
C	Cost, Eq. 4.2
c_k	Cost at time step k, Eq. 4.2
d	Probability density function, Eq. 2.6
$\mathbb{E}$	Expected Value
$\hat{f}$	Generic function approximation
f	Generic function
f_k	Evolution function for a dynamic system at time step k, Eq. 4.1
F_r	Froude number
g	Gravity acceleration
G_M	Metacentric height
G_Z	Arm of hydrostatic moment
h	Heel angle
H_w	Significant wave height
h_{rad}	Heel angle in radians
I_{yy}	Mass moment of inertia
k_{yy}	Longitudinal radius of gyration
$\hat{\ell}$	Log-likelihood function, Eq. 2.9
$\mathscr{L}$	Likelihood function, Eq. 2.7
L	Lift
L_{CB}	Longitudinal centre of buoyancy, positive backwards
L_{CF}	Longitudinal centre of waterplan area, positive backwards
L_{wl}	Length of the waterline
N	Number of steps for DP algorithm, Eq. 4.2
(p,q)	Parameters defining ARMA model
$\mathbb{P}$	Probability
$P = P_{ij}$	MC transition matrix
p^0	MC Initial distribution
R	Error function for SVR, Eq. 2.13
R^2	Coefficient of determination
Re	Reynolds number
S_w	Wetted surface area of the hull
S_{wc}	Wetted surface of the canoe body
$\mathscr{T}$	Test set, Eq. 5.4

T Draft of the hull
T Time, Eq. 4.3
t Time
T_c Draft of the canoe body from waterline
T_d Time difference between two racing yachts
T_{switch} Time for switch between different risk attitudes
Th_{app} Thickness of the appendage
$\mathscr{U}_k$ Admissible DP controls at time step k
u_k^{opt} Optimal decision at time step k, Eq. 4.1
u_k Decision at time step k, Eq. 4.1
$\mathbf{w}(t)$ Wind velocity at time t
w_i, w_j States of MC
wd Wind direction
ws Wind speed
w_i Synaptic weights, Eq. 2.11
$\bar{x}$ Input vector for SVM, Eq. 2.12
x_k State of a dynamic system at time step k, Eq. 4.1
$\mathbf{y}(t)$ Generic input for wind forecasting at time t, Eq. 2.1
Z_C Arm for the aerodynamic side force
Z_k Distance from the centre of buoyancy of the keel to the bottom of the hull
Z_{app} Arm for the appendages forces
z_i Random variables uniform in [0,1]
AC America's Cup
ANN Artificial Neural Network
ARMA Autoregressive Moving Average
BS Boat Speed
BS* Corrected Boat Speed
CFD Computational Fluid Dynamics
DP Dynamic Programming
EI Effectiveness Index
MAE Mean Average Error
MC Markov Chain
MEI Mean Effectiveness Index
MLE Maximum likelihood estimator
MLP Multi-Layer Perceptron
MSE Mean Square Error
NWP Numerical Weather Prediciton
PCC Pearson's Correlation Coefficient
RMP Race Modelling Program
SVM Support Vector Machines
SVR Support Vector Regression

TWA	True Wind Angle
VMG	Velocity Made Good
VPP	Velocity Prediction Program
α_i, α_i^*	Solution of SVR quadratic problem, Eq. 2.16
ε_t	Forecast error at time t, Eq 2.2
ω_k	Random variable influencing dynamic system at time step k, Eq. 4.1
Φ	Mapping function for SVM, Eq 2.12
ψ	Activation function within a neuron, Eq. 2.11
τ	Tacking penalty
θ	Vector of parameters defining a model, Eq. 2.6
ξ_i	Independent identically distributed random variables, Eq. 2.5
ζ	Weighted sum computed within a neuron, Eq. 2.11

Chapter 1

Introduction

A yacht race is a competition where two or more boats race each other to complete a certain course in the shortest possible time. Races are held in many different formats and levels: in the case of *match race* only two boats face each other, while in a *fleet race* the number of participants can be very high. One of the most prestigious sailing competition (and by far the oldest and most expensive) is the America's Cup, which includes match races between various teams fighting for the chance of challenging the Cup defender, i.e. the winner of the previous edition.

Usually, a race course includes several turns around an upwind and a downwind mark, where the marks are aligned with the wind. The course is designed to present some challenges to the skippers. In fact, the speed of a sailing yacht depends on the wind speed and the course wind angle (TWA, the supplementary angle between the wind velocity and the boat heading). Figure 1.1 presents an example of boat speed (BS) as a function of the TWA for a given wind speed in a polar diagram. A polar plot of this kind, which may include different curves associated to different wind speeds, is the conventional way of presenting the boat speed, and although the actual BS can depend on other factors (such as waves and crew), it is considered as a characteristic of a yacht. As shown in the plot, the highest values for the BS are achieved when

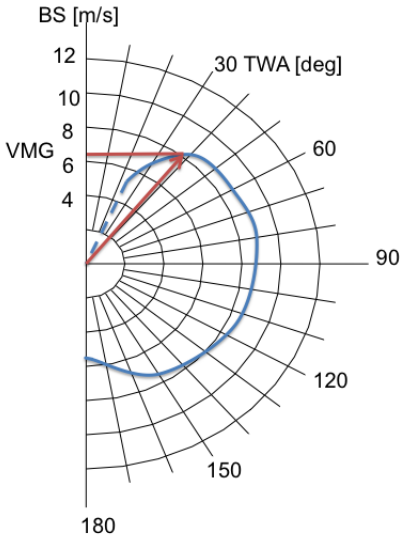

Figure 1.1: Example of polar diagram.

sailing at a TWA of approximately 90° (on a *beam reach*). Conversely, when the TWA tends to zero, BS tends to zero. Therefore, when sailing upwind (for instance, from a downwind mark to an upwind mark), the most effective route consists in a zig-zag in the wind direction, sailing at a TWA of 35°-50° (*close hauled*). In this case, a skipper's aim is to maximise the speed in the upwind direction, which means to find the TWA such that the projection of the boat velocity on the upwind direction is a maximum. The corresponding velocity is referred to as Velocity Made Good (VMG) and is shown in red in Figure 1.1. Similarly, a VMG can be defined for downwind sailing as the projection of the boat velocity on the downwind direction.

The VMG can be defined also for downwind sailing. In fact, as shown in Figure 1.1, even if the

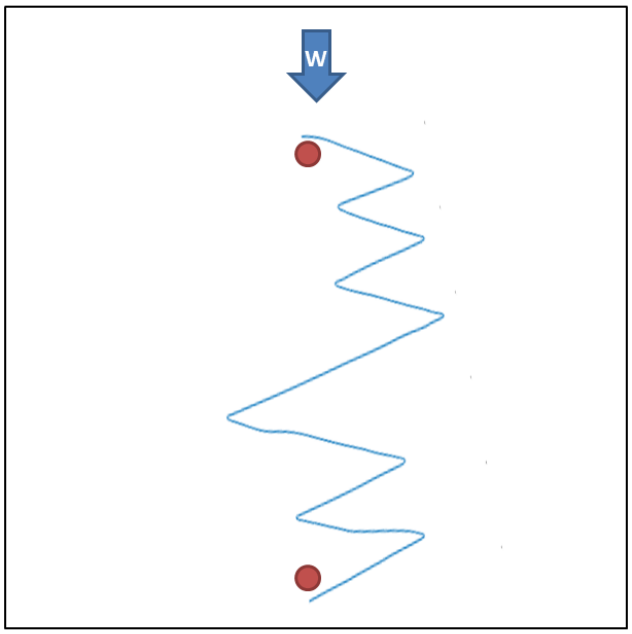

Figure 1.2: Example of upwind leg.

velocity is not null when the TWA is 180°, the maximum projection on the downwind direction for this example is obtained at angles of approximately 150°. However, the optima angle for downwind sailing can have significant variations depending on the yacht geometry.

A typical upwind course is shown in Figure 1.2, where an example of an upwind leg of one of the races of the 34^{th} America's Cup finals is plotted. The arrow represents the average wind direction, and the yacht (Emirates Team New Zealand in this example) is sailing upwind. The changes of direction in the zigzag are called *tacks*. For the downwind case, the changes of direction corresponding to tacks are called *gybes* (or jibes). During a tack, the yacht is not sailing at its optimal VMG, therefore each tack leads to a time loss, which again is dependent on the kind of yacht considered. A somewhat trivial but important aspect also shown in Fig. 1.2 is that the yacht is never sailing on a straight line. This depends on a number of factors: current changes, waves, manoeuvring and crew movement, but most importantly wind changes. If all

of these factors were neglected, the optimal course would be the one including just one tack halfway through the upwind leg. In practice, sailing conditions continuously change. Moreover, the racing area is often limited, forcing the inclusion of more tacks.

1.1 Yacht racing tactics and strategy

The terms *tactics* and *strategy* are often used as synonyms to indicate the set of decisions taken by a skipper with the aim of winning the race. However, they can be used to identify two different problems that a skipper has to face. Strategy refers to the positioning of the yacht on the course, while tactics refers to the position of the yacht with respect to the opponent. Also *Strategy involves the big picture; tactics focusses in close. Strategy is long term and planned, tactics is more immediate and spontaneous* (Gladstone, 2002).

The difficulty of the sailing decision-making process can be understood by noticing that in many competitions it is customary to have more than one person involved in such a process. Although the crew roles change depending on the kind of boat and competition, an interesting example is given by the following description offered in a BBC Sports article on the 31^{st} America's Cup (BBC, 2002):

Skipper: Ultimate responsibility for boat's safety and performance as well as the public face of the team off the water. Positioned in the afterguard or "brains trust" and usually doubles up as helmsman, tactician, or strategist.

Helmsman: Concentrates on driving the boat as fast as possible. Takes input from tactician, navigator and trimmers.(...)

Tactician: Responsible for the positioning of the boat on the course, taking into account wind, tide, sea conditions and the state of the race. Draws on input from the crew, strategist and navigator and must have implicit trust of the helmsman, especially during the frantic pre-start. It is a major role and the tactician acts as the eyes of the boat.

Navigator: In charge of all the electronic instruments on board, which provide information on conditions, boat performance and meteorological input, often gathered months before the event.

Strategist: A diverse role. A key member of afterguard who is on a constant lookout for changes in wind direction or strength.The strategist also helps trim the mainsail (...) and he acts as an extra pair of hands during manoeuvres.

This example doesn't represent a strict classification, but it underlines the complicated nature of the decision-making process during a race, that in this case is shared between up to five crew members. Things have changed since the 31^{st} America's Cup, and the event is now at its 35^{th} edition. Boats have become the much faster foiling catamarans, which are physically demanding and are sailed by only six crew members. Everything happens faster, there isn't a person dedicated solely to tactics or to strategy, but the decisions must be taken while accomplishing also some physically demanding tasks.

The term *strategy* can also have a more precisely defined mathematical meaning in the framework of Dynamic Programming, as a synonym of policy, defined in Chapter 7. In this thesis all the decisions generated as a result of Dynamic Programming will be referred to as policy, while the general distinction between tactics and strategy remains valid. The need of building a strategy based on anticipated wind shifts, rather than just reacting to the observed wind changes, can be explained with a simplified example, presented in Tagliaferri *et al.* (2014). Figure 1.3 shows a schematic representation of the route followed by two boats in order to reach an upwind mark in two different wind conditions. In the first case (left) the wind alternately shifts by 3° to the right and to the left. The first shift is towards the right and, while the two boats sail at their optimum course wind angles, they chose to sail in different directions. Both boats tack at every wind shift. The black boat is always sailing towards the left hand side of the race course when the wind shift is towards the right, and vice versa when the wind shift is towards the left. Conversely, the grey boat has the opposite strategy. Therefore, the black boat is always sailing to a closer angle to the mark than the grey boat, she sails a shorter course and she arrives first to the mark. The winning strategy of the black boat is that she always maximises her velocity towards the mark. However, this is not always a winning tactic. In fact, in the second case (right), the wind constantly shifts towards the right. The two boats follow the same strategy as in the previous case: experiencing a wind shift to the right, the black boat sails towards the left and the grey boat sails towards the right. As the wind is constantly shifting to the right, the two boats never tack until they reach a lay line, i.e. where a tack allows the mark to be reached without any further tacks. The black boat needs to sail most of the race course before reaching the lay line and being able to tack to the mark, while the grey boat reaches the lay line before the black boat. In this case, even if the two boats have pursued the same strategy based on the wind observed at the time, the resulting course sailed by the black boat is longer than the course sailed by the grey boat, because the wind shifted regularly in the same direction instead of alternating to opposite directions. This shows that the decision cannot be based on the wind direction observed at the time, but that the future wind shifts must be foreseen in order to develop a winning strategy.

Another important factor that a skipper has to consider in her decision-making process is the presence of one or more opponents. The tactical interaction between competing yachts is regulated by the internationally recognised ISAF Racing Rules (ISAF - International Sailing Federation, 2013-2016). Those rules are aimed at guaranteeing the safety of races, for instance by defining clearly, when boats meet in any situation, which one has right of way and which one must keep clear of the other. As a consequence, they create room to play.

The presence of an opponent has some physical and tactical consequences together with changing the structure of the problem itself. In fact, a skipper knows that her aim is not to complete the race in the shortest possible time, but to finish it *before* the opponent(s). One of the scientific questions that will be investigated in this work is the following:

Does a strategy aimed at minimising the expected time to complete a race lead to the maximum probability of winning a match race?

This can be investigated in terms of *risk attitude*, exploring whether the intuitive common

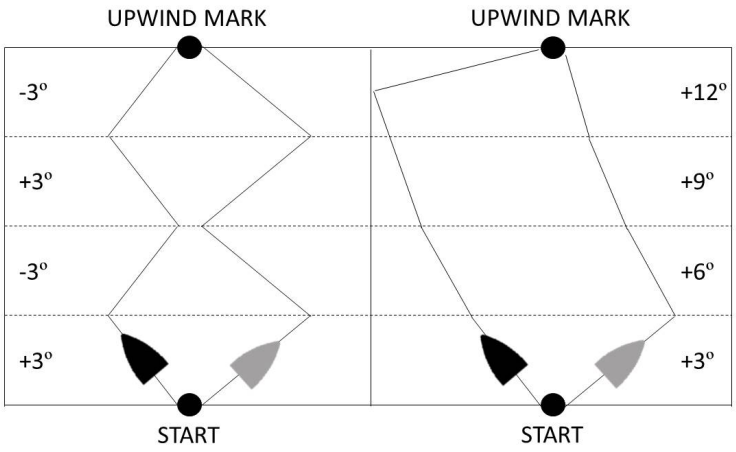

Figure 1.3: Example of upwind leg, appeared in Tagliaferri *et al.* (2014).

behaviour of "when losing, risk, when winning, be conservative" can be translated in a higher probability of winning.
All the factors (future wind changes, sea state, yacht's physics, racing rules, opponent's behaviour) need to be taken into account by a sailor when taking decisions during a race. A skipper continuously collects information on the boat performance and the sailing environment, makes predictions, and, ultimately, takes a decision on the course of actions to follow.
This leads to a broad engineering question addressed in this thesis: is there a way to efficiently help a skipper/tactician on board in his decision-making process, combining optimal tactics and strategy?

1.2 Literature review on yacht racing

A scientific approach to the problem of yacht racing strategy has been developed only in recent years. In fact, previous research related to competitive sailing was mainly focussed on understanding a yacht's dynamics, in order to improve the design process and create faster and more efficient boats. The outcome of this research was the development of Velocity Prediction Programs (VPP), computer programs that solve the equations of motion for a sailing yacht, and determine its velocity for given wind conditions. Kerwin and Newman (1979) presenst one of the earliest studies detailing the development of a VPP. The introduction of such tools determined a big step forward for competitions such as the America's cup. In fact, United States had always successfully defended the trophy since its creation, in 1851, but in 1983, for the first time, an Australian team managed to win, and this success was mostly due to the radical innovation in keel design for the yacht *Australia II*, (Oossanen and Joubert, 1986). Americans learned from this defeat, and the following campaign saw a massive effort in applying state-of-the-art technology to the design of the yacht that would challenge the Australians. The resulting yacht, *Stars and Stripes*, succeeded in the mission of bringing the Cup back to the US in

1987, after a campaign that was the first to see a competition not only between sailors but also between the engineering teams of the different countries. This aspect is passionately described in the paper *Stars and Stripes* by Letcher *et al.* (1987) which focuses on the advantages brought by computer technology.

Since then, the competition has evolved along those lines, and today it is still a fierce battle between engineering teams besides sailing teams, and in sports journalism America's Cup races are often compared to Formula One GP. The development of VPP allowed design teams to compare different design choices at an early stage, and techniques used for determining forces acting on the boats are determined using a variety of techniques, both experimental and numerical.

Letcher recognised that *yacht racing has an essentially random component in that the relative performance of two yachts depends on the wind speed and the sea conditions, which vary randomly from day to day. VPP results by themselves are therefore inconclusive and possibly misleading for determining the order of merit of two candidate yachts over a series of races. (Letcher* et al.*, 1987)*

The evolution of VPP lead to *Race Modelling Programs* (RMP), computer programs aimed at simulating an entire race between two yachts. The *Stars & Stripes* campaign involved one of the very first RMP to analyse the probabilities of win/loss of a yacht.

The subsequent America's Cup saw the development of a RMP which included a statistical weather model based on site-specific environmental data for San Diego. This RMP was developed by the Partnership for America's Cup Technology, and details are described by Gretzky and Marshall (1993).

In those models, the tactical decision process is modeled as a set of fixed decision rules. The tactical and physical interactions between the yachts are not adequately modeled, and this limitation is reflected in the definition of win in Letcher *et al.* (1987), where a yacht has to win by a certain time margin to be certain of a win.

An important contribution to RMP came from the studies carried out at The University of Auckland in collaboration with the New Zealand challenger team. Philpott *et al.* (2004) developed a model to predict the outcome of a match race between two competing designs, still assuming a set of fixed decision rules but taking into account some interactions between yachts (for instance, when crossing). Philpott and Mason (2001) investigated the decision-making process, focussing on the development of a strategy. In this work, which constitutes a fundamental basis for the present study, dynamic programming is used to generate a *policy*, that can be computed before the race, and can then be used during the race. More detail on this work is given in Section 4.5.

In these two studies the tactics and strategy modeling was aimed at obtaining a simulation tool that could replicate as closely as possible the situations that can arise during a yacht race, with the ultimate objective of assessing competing designs. Other studies not directly related to the America's Cup have tackled the problem of decision-making for sailors. Ferguson and Elinas (2011) propose a simple Markov decision model, where at all times the sailor has only two options, "do nothing" or "tack". The work developed in this study, focussed on inshore racing,

includes a VPP and a model for wind flow around landmasses. The importance of the tacking penalty is investigated by comparing routes produced by assuming different penalty factors associated to tacks.

Recently, the University of Southampton has developed a sailing simulator called "Robo-Race", a tool to model both the physical behaviour of a yacht and the interaction with the crew (Scarponi *et al.*, 2007a,b). The tool is designed so that human sailors can interact with it, racing against a computer in an artificial environment. A VPP using four degrees of freedom is implemented, including the tacking model based on the studies of Masuyama *et al.* (1995). Improvements on the first implementation are focused on physical interactions between racing yachts (Spenkuch *et al.*, 2008, 2011), and the dynamics of the yacht during manoeuvring (Banks *et al.*, 2010; Spenkuch *et al.*, 2010), both for upwind and downwind sailing. An important contribution of this work is recognising the existence of conflicts between strategy and tactics, for instance when a yacht decides to tack to avoid the blanketing effect from another yacht, but doing so it incurs in an unfavourable wind.

In Kimball and Story (1998) optimal sailing trajectories are described by using Fermat's principle, which states that light takes the path of minimal (extremal) time. The authors give an elegant example of how Huygens' principle (the geometric construction which describes the paths of light rays) can be "borrowed" from optical science to characterise minimum-time paths of sailing yachts. The approach in this work is purely theoretical, and not aimed at building a decision tool. However, it underlines the challenging nature of the sailing strategy problem, and the suitability of advanced mathematical tools to tackle it.

1.3 Problem statement

This thesis sets out to investigate the possibility of computing a sailing strategy for a match race that can defeat an opponent whose strategy is to minimise the expected time of completion of the race.

This is accomplished by developing a tool that emulates the decision-making process during a yacht race, in particular an inshore match race. This tool must be able to emulate and improve the human decision process. The proposed algorithm must receive as input all the set of information that is usually available to a sailor, obtained from on-board instrumentation, from visual observation, or because it has been previously computed (for instance, the boat's polars). In particular, the following quantities are assumed to be known:

- Boat position - the boat coordinates and velocity on the race course are assumed to be known at any time.
- Boat's polars - BS is assumed to be known as a function of TWS and TWA. A methodology for developing a simple VPP that takes as input the boat's geometry to compute BS is outlined in Section 6.
- Sea conditions - the tidal current is assumed to be a known function of time and position on the race area. Waves and other currents are assumed negligible.

- Wind forecast - a long-term wind forecast is usually available to sailors before the race, and this information can be used by the proposed algorithm.
- Wind history - wind speed and direction can be recorded on the location of the race before and during the race.
- Opponent's position.
- Crew/yacht performance - an estimate of the time losses associated to manoeuvrings such as tacks.

In order to generate the optimal decisions for a skipper, the following objectives must be accomplished:

- Integrating the wind forecast obtained from external sources before the races with a shorter-term forecast generated in real-time and based on the wind observed while racing.
- Optimising the methodology for generating the shorter-term forecast, based on the available information.
- Including a model for the competitor, that allows to consider tactical decision together with strategic decisions.
- Computing the optimal decisions in a computationally efficient way, so that the algorithm can be implemented in a computer program to be used on board during races.

1.4 Thesis outline

After the present Introduction, this thesis is divided in four major parts:

1. Background
2. Method
3. Results
4. Concluding Remarks

The Background and the Method have the same structure and are divided in three chapters which address the three major topics of this work:

- Wind forecasting
- Velocity prediction and aerodynamic interactions between two boats
- Routing algorithms

The chapters in the Background part aim at giving to the reader the basic concepts necessary to understand the Method, and can be skipped without loss of continuity by the expert reader.
The results are presented in three chapters as well: first the results on the wind forecasting are presented, then the optimum routes for a boat in isolation based on these forecasts and, finally, how the optimum route varies due to the presence of a competitor is discussed.
Conclusions and Future work constitute the two final Chapters.

1.5 Publications

Significant parts of the work outlined in this thesis have been published in the following academic papers:

- Tagliaferri F., Philpott A.B., Viola I.M., Flay R.G.J. On Risk Attitude and Optimal Yacht Routing Tactics, *Ocean Engineering*, Volume 90, 1 November 2014, pp. 149-154
- Tagliaferri F., Viola I.M. Wind Direction Forecasting with Artificial Neural Networks, *Ocean Engineering* Volume 97, 15 March 2015, Pages 65-73
- Tagliaferri F., Hayes B., Viola I.M., Djokic S., Wind Modeling with Nested Markov Chains, submitted to *Wind Energy* (20/11/2014)
- Tagliaferri F., Philpott A.B., Viola I.M., Flay R.G.J. Optimal Yacht Routing Tactics, *3rd International Conference on Innovation in High Performance Sailing Yachts (INNOV'SAIL)*, 26-28 June 2013, Lorient, France
- Tagliaferri F., Viola I.M., Dow R.S., Artificial Neural Networks Wind Forecast for Safety at Sea and Yacht Racing Tactics, *17th International Conference on Ships and Shipping Research (NAV2012)*, 17-19 October 2012, Naples, Italy
- Tagliaferri F., Viola I.M. Feedforward Neural Networks for Very Short Term Wind Speed Forecasting, *Marine and Offshore Renewable Energy*, 26-27 September 2012, London, UK

The most relevant publications are attached at the end of this Thesis.

PART I

Background

Chapter 2

Background on wind forecasting

This chapter is aimed at presenting introductory concepts on wind forecasting focusing on the very-short time horizon which is of interest to sailors competing in inshore races. Comprehensive reviews on the state of the art in wind forecasting include Soman *et al.* (2010) and Jung and Broadwater (2014).

In the previous Chapter it was argued how a successful sailing strategy is based on the correct anticipation of future wind behaviour. Sailors must take advantage of all the information available up to the point in time when the decision is taken. Sailors have access prior to the race to medium and long term wind forecasts from reliable sources. However, forecasting ability is required by the sailor to address very-short-term wind changes. Moreover, there are often some local characteristics that constitute an advantage for sailors that are familiar with the race's location. The forecasting module developed in this work is aimed at improving the forecasting ability of the sailor. Its objective is to generate short-term wind forecast to correct the long-term trend obtained from an "external" forecast, similarly to what a human sailor does.

The methodology chosen to develop this module is based on Artificial Neural Networks (ANN), because besides their proven ability to model highly non-linear functions, they are suitable for this problem because of their ability to emulate the human peculiarity of "learning from experience".

In recent years, the research on wind forecasting has been driven mainly by the need of accurate forecasting for wind energy production. It is not surprising then that most of the literature on this topic is related to wind farm applications, as is its terminology.

A first major classification for wind forecasting models is based on the time range of the forecast. This distinction is not universally nor strictly defined, and it can depend on the particular application and use of the forecast. In Soman *et al.* (2010) time ranges are divided in four groups, inspired by applications in wind power generation. These groups are described in Table 2.1, which shows a comparison between the terminology used in most wind energy-related literature, and the terminology adopted in this work, which is aimed at describing wind forecasting from a sailor's perspective. The classification of different time ranges is shown in Columns 1 and 2, while Column 3 shows examples of applications in the field of operation of electricity systems. Column 4 gives some examples of use of wind forecasts by sailors before and during an inshore race, and common forecast sources for sailors are listed in Column 5.

A second major distinction in wind forecasting models (Jung and Broadwater, 2014) is whether

Table 2.1: Forecasting time horizons in different applications

Time horizon	Range	Applications in wind energy	Applications in sailing	Common sources for sailors
Very short-term	Seconds to minutes ahead	- Electricity market clearing - Regulations actions	-Sails/rudder regulations - Tacking/jibing decisions	Visual observation of water surface and of heading of the other boat(s)
Short-term	Minutes to hours ahead	- Economic load dispatch planning - Load Increment/ Decrement decisions	- Preferred side of the course - Sails hoisting - Strategy	NWP-based wind forecast
Medium-term	Up to one day ahead	- Generator online/ offline decisions - Operational security in day-ahead electricity market	- Mast regulations - Choice of sails to be carried onboard	NWP-based wind forecast
Long-term	More than one day ahead	- Unit commitment decisions - Maintainance scheduling to obtain optimal operating cost	- Boat maintenance - Training calendar	NWP-based wind forecast

the physical information of the location of interest is used as input for the model. This information can include quantitative and qualitative data for terrain roughness and orography. This leads to a physical approach called *Numerical Weather Prediction* (NWP). Conversely, if the forecast is based on past values of wind velocity (and possibly of other quantities such as pressure, temperature, humidity etc.) it is commonly referred to as *statistical forecast.*

More methods for forecasting have been proposed, that don't necessarily belong to one of those two categories. Most of them are of very recent development. Similarly to the statistical models, these models do not include any physical assumption about the reality that they aim to model. Some successful examples include Grey predictors (El-Fouly *et al.*, 2006), which relate future values of the wind time series to past values in terms of a differential equation, Support Vector Machines (Du *et al.*, 2008), and other data mining techniques (Zheng and Kusiak, 2009).

ANN can be classified as a statistical method. However, due to their wide use and the many extensions of the basic model, they are often classified as a third, independent class of forecasting techniques (Soman *et al.*, 2010).

In the following paragraphs, the most used methods for wind forecasting are described.

2.1 Numerical methods

NWP is based on the numerical solution of conservation laws in the Earth's atmospheric boundary layer, and therefore all NWP models are conceptually based on Computational Fluid Dynamics (CFD). However, models differ from each other in the way the terrain and atmosphere are discretised, and in the way the equations are numerically solved. Successful examples include the models *Predictor*, developed by Rise National Laboratory in Denmark (Landberg, 1999, 2001), *Previento*, developed by the German University of Oldenburg (Focken *et al.*, 2001), and *LocalPred*, developed by the Spanish CENER (Marti *et al.*, 2003). Generally, NWP is expensive in terms of computational resources, time, and implementation. As a consequence, those methods are used mostly for predictions for the medium and long term.
NWP outputs are commonly used by sailors, who usually access weather bulletins from days in advance up to the beginning of the race, to gather information on the general expected behaviour of the wind. For instance, the wind conditions expected during the race determines initial tension in the mast rigging, the sails to be carried on board, etc.

2.2 Statistical methods

Statistical forecasting is based on the assumption that future wind velocities can be described as a function of past wind velocities, and in some cases of past values of other quantities (e.g. temperature, pressure, etc), as shown in Equation 2.1.

$$\mathbf{w}(t) = f(\mathbf{w}(t-1), \mathbf{y}(t-1), \mathbf{w}(t-2), \mathbf{y}(t-2), \dots) \tag{2.1}$$

where $\mathbf{w}(t)$ represents the wind velocity vector at time step t, and $\mathbf{y}(t)$ is a generic vector containing other input values for the discrete time t. The aim of a statistical forecasting technique is to approximate the function f, by finding a function $\hat{f}$ such that

$$\mathbf{w}(t) = \hat{f}(\mathbf{w}(t-1), \mathbf{y}(t-1), \mathbf{w}(t-2), \mathbf{y}(t-2), \dots, \mathbf{w}(t-m), \mathbf{y}(t-m)) + \varepsilon_t \tag{2.2}$$

where ε_t represents an error which in the following will be omitted. In the following, the symbols with the hat are used to represent estimated or approximated quantities.

2.2.1 Persistence model

The simplest possible method is the so-called *persistence model*, based on the assumption that the wind at time step t will be identical to the wind at time step $t-1$. For the persistence model Equation 2.2 becomes:

$$\mathbf{w}(t) = \mathbf{w}(t-1) \tag{2.3}$$

The average error of the persistence model corresponds therefore to the average difference of consecutive values in the wind time series. Besides its triviality, this method shows a relatively good performance, especially for short and very short term forecasting (Milligan *et al.*, 2003). In fact, it is commonly used as a benchmark for statistical methods. In fact, a "low" absolute error is significant only if it is lower than the corresponding persistence error. As will be discussed in Section 8.4.1, this is particularly important in very-short-term forecasting.
A comparison with the persistence model provides an answer to the question "is error small because the forecasting model is good, or because the wind variations are small?". For this reason, it is not uncommon to find in the literature studies that present the forecasting error as a percentage improvement over persistence model, as done, for instance, in Alexiadis *et al.* (1999).

2.2.2 Autoregressive Moving Average processes

Often, the function f in Equation 2.1 is approximated by a function $\hat{f}$ which is assumed linear. The value $\mathbf{w}(t)$ is then a combination of an arbitrary number of previous values. This leads to *autoregressive* models, and their generic formulation is

$$\mathbf{w}(t) = \sum_{i=1}^{p} a_i \mathbf{w}(t-i) \tag{2.4}$$

This model is referred to as AR(p) model in the literature, and is widely used for time series analysis, generation of artificial time series and also forecasting. It is possible to generalise AR models by adding a moving average term, which is a linear combination of independent identically distributed random variables (noise terms). This constitutes the more generic ARMA model. Those models have been extensively studied and are the basis of time series analysis. A detailed description of those models can be found, for instance, in Hamilton (1994). The generic formulation for an ARMA process is

$$\mathbf{w}(t) = \sum_{i=1}^{p} a_i \mathbf{w}(t-i) + \sum_{i=1}^{q} b_i \xi_{t-i} + c \tag{2.5}$$

where p and q are the indices defining the model, which is usually noted as ARMA(p,q), and ξ_i are the independent random variables. If all $b_i = 0$, then the model is again the autoregressive model AR(p). Conversely, if all the a_i are null, the process is a moving average, MA(q). ARMA can be used to perform a forecast of the time series $\mathbf{w}$ by using historical data to find the coefficients a_i, b_i, c.
There are standard techniques for computing the coefficients a_i and b_i for an optimal fitting of the dataset. The one that is used this work is the *maximum likelihood*, which allows to estimate the parameters of a model so that the model becomes the one that is "most likely" to have generated the data.
Let us consider an independent and identically distributed sample. The generic joint density

function d for the sample is defined as

$$d(x_1, x_2, \ldots, x_n \mid \theta) = d(x_1|\theta)d(x_2|\theta)\cdots d(x_n|\theta). \tag{2.6}$$

where θ represents the vector of parameters defining the model. For instance, for an ARMA process, the vector θ contains the values p, q, a_i, b_i, c. The symbol $\cdot|\cdot$ in the conventional probability notation represents a conditional value. Let us now consider the observed values $x_1, x_2, \ldots, x_n$ to be fixed "parameters" of this function, whereas θ is the function's variable and allowed to vary freely; this function is called *likelihood function*:

$$\mathscr{L}(\theta\,; x_1, \ldots, x_n) = d(x_1, x_2, \ldots, x_n \mid \theta) = \prod_{i=1}^{n} d(x_i|\theta). \tag{2.7}$$

In practice it is often more convenient to work with the logarithm of the likelihood function, called the log-likelihood:

$$\ln \mathscr{L}(\theta\,; x_1, \ldots, x_n) = \sum_{i=1}^{n} \ln d(x_i|\theta), \tag{2.8}$$

or the average log-likelihood:

$$\hat{\ell} = \frac{1}{n} \ln \mathscr{L}. \tag{2.9}$$

Indeed, $\hat{\ell}$ estimates the expected log-likelihood of a single observation in the model.
The method of maximum likelihood estimates θ_0 by finding a value of θ that maximises $\hat{\ell}(\theta; x)$. This method of estimation defines a maximum-likelihood estimator (MLE) of θ_0

$$\{\hat{\theta}_{\text{mle}}\} \subseteq \{\arg\max_{\theta \in \Theta} \hat{\ell}(\theta\,; x_1, \ldots, x_n)\}. \tag{2.10}$$

In practice, using the maximum likelihood function to estimate a parameter answers the following question: assuming that the data has been generated by a fixed model, which depends on a parameter θ, what value of θ assigns the greatest probability to the generated dataset? When possible, the use of a MLE for the models parameters avoids the lengthy procedure of trial-and-error. Unfortunately, it is not always possible to explicitly define the log-likelihood function. More details on parameter estimation are given in Chapter 5.

2.3 Artificial Neural Netwoks

Models where the function f defined in Equation 2.1 is approximated with a function $\hat{f}$ which is not linear are also possible. ANN belong to this class. In fact, an ANN models a function as a linear combination of nonlinear functions. Details on ANN will be given in the next Section. In Section 2.3 an introduction to ANN and their basic terminology will be given, before presenting the most important results in wind forecasting literature that lead to the choice of ANN for very short term wind speed and direction forecasting used in this work.

ANN are computational models inspired by the human brain. In particular, they are aimed at emulating the brain's capability to learn from experience and to fire a high number of connections between neurons in a very short time.
In the literature it is possible to find a vast number of applications where ANNs have been successfully used, especially to solve problems which are peculiar of humans, such as speech recognition (Morgan and Bourlard, 1995), image classification (Park *et al.*, 2004), control of moving robots (Fierro and Lewis, 1998).

The Neuron

ANN are highly parallel computational models, that combine many single processing units called neurons. Each neuron takes several inputs originating from other neurons, and produces an output that is then transmitted to other neurons. A representation of a neuron's structure is shown in Figure 2.1. It can be broken down into the following components:

1. A set of connecting links, called synapses, where the i-th synapse is characterised by a weight w_i (synaptic weight);
2. An adder within the neuron that sums each i^{th} input multiplied by weight w_i;
3. An activation function, ψ, which transforms the sum computed by the adder into the neuron output y. If the activation function is linear, a neuron results in a linear combination of the input values, while non-linear activation functions allow for the modelling of non-linear problems.

Therefore, a neuron receiving an input vector $x = (x_1, \ldots, x_n)$ and producing output y can mathematically be described by Equation (2.11):

$$y = \psi(\zeta); \zeta = \sum_{i=1}^{u} w_i x_i \tag{2.11}$$

ANN structures and learning processes

Neurons can be assembled together in different ways. The simplest, most used structure is the Multi-Layer-Perceptron (MLP). MLP are characterised by a unidirectional flow of information, and this property is referred to as *feed-forward*. This means that each neuron processes the information only once. Moreover, neurons in a MLP are organised in *layers*, and neurons belonging to the same layer do not communicate between each other, while the output of the neurons of one layer becomes the input of the neurons in the following layer. Two layers communicate with the 'external environment': the first layer, that receives the external input, called *input layer*, and the last layer, transmitting the network output back to the external environment, called *output layer*. In most cases, the input layer has also the function of scaling and normalising the external input, and this process is reverted by the output layer. All the other layers between the input and the output layer are called *hidden layers*. Figure 2.2 shows

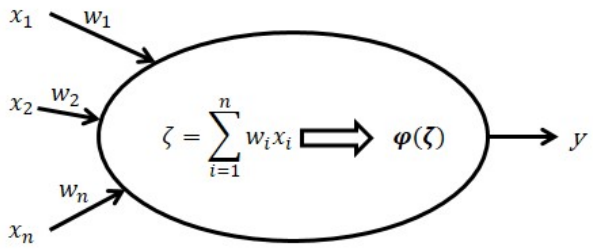

Figure 2.1: Schematic representation of a neuron - appeared in Tagliaferri *et al.* (2015).

an example of a MLP with two hidden layers, taking as input the vector of $m+1$ previous wind velocity values and computing a one-step-ahead forecast.

The success of the MLP is largely due to its property of constituting a class of universal

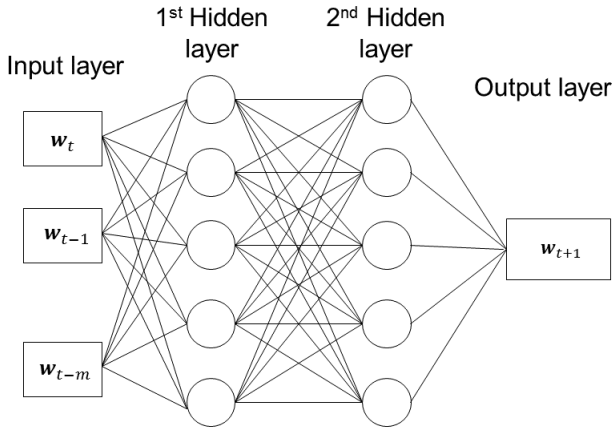

Figure 2.2: Example of multi-layer perceptron - appeared in Tagliaferri *et al.* (2015).

approximators of nonlinear function. In fact, Hornik proved in 1989 that *standard multilayer feedforward networks with as few as one hidden layer using arbitrary squashing functions are capable of approximating any Borel measurable function from one finite dimensional space to another to any desired degree of accuracy, provided sufficiently many hidden units are available* (Hornik *et al.*, 1989). This means that for a wide class of functions, including all continuous functions (and a wide range of "reasonable" non-continuous functions, see Kolmogorov and Fomin, 2012, for details), it is possible to build a multilayer perceptron that models the function so that the difference between the function values and the MLP-computed values is less than

any given threshold. However, in practice this might not always be applicable due to lack of training data, presence of non-deterministic relationship between input and output, or simply because the required number of neurons is too high. The number of neurons required can be decreased by using more than one hidden layer.

The learning process, aimed at making the network able to model a specific problem, involves the continuous modification of the synaptic weights. A commonly used training algorithm is based on the principle of iterative error-correction. The synaptic weights of the various neurons are initialised to random values, then a training set of input and output data is presented to the network. For each input vector, the initially generated output vector is compared with the known true output vector. The synaptic weights of the output layer are then modified by adding a factor that is proportional to the error and to a learning rate, and those corrections are extended to all of the weights in the network through a back-propagation process up to the input layer. This operation is iterated until successive changes in the synaptic weights are smaller than a given value, or when the errors begin to increase. For further details on training algorithms and validation processes, see Haykin (1994).

Other classes of ANN have been proposed. Some structures are variation of the MLP, while others present loops and are classified as recurrent structures, often still organised in layers. Self organising maps are networks organised in layers, where instead of modifying the weights and averaging the output, neurons are selected using a competitive learning (Kohonen, 1990) . These models are mainly used for classification problems. Hopfield networks (Hopfield, 1982) are single-layer recurrent networks, trained with an algorithm to learn to recognise patterns by echoing them back in the network. Radial Basis Function networks (Renals, 1989) are feed-forward multilayer networks where the hidden layer is based on a radial basis function. They are used to model more complex activation function than the MLP. Elman recurrent networks are structures used successfully in forecasting, where a context layer holds the output of the previous iteration, allowing for multi-step prediction (Elman, 1991). They are generally more computationally demanding than MLP.

2.3.1 Literature review on ANN in forecasting

ANN have been used in many studies for wind speed forecasting, starting to gain popularity at the end of the past century. Different structures and different training algorithms have been tested. The studies are mostly application-driven, therefore they are developed by looking for the optimal ANN structure in a chosen ANN class, for a time range that depends on data available.

In one of the earliest studies (Mohandes *et al.*, 1998) ANNs are compared to AR processes for forecasting daily and monthly averages. Surprisingly, the optimal AR process for the used data set is found to be AR(1), and this model is outperformed by the ANN proposed. In Alexiadis *et al.* (1999) ANN are used to forecast wind speed from 15 up to 180 minutes in advance. The peculiarity of this study is in the use of input data coming from different locations than the one for which the output is needed (specifically, some wind stations located upwind with respect to

the output station at up to 40km of distance). Different input combinations are tested, showing an improvement of 20%-40% compared to the persistence model.

Although these early studies suggest that a small number of inputs are needed in order to obtain a reasonable forecast, some other studies show that the use of other input quantities such as air temperature, humidity and pressure can improve the forecasting ability of ANNs for the short-medium term, as shown by Hayashi and Kermanshahi (2001) and for the short-very short term as shown by Pérez-Llera *et al.* (2002), and more recently Welch *et al.* (2009).

The improvement over more refined linear models than the ones by Mohandes *et al.* (1998) is presented in More and Deo (2003), where different ANN are compared with ARMA models for long term forecasting. It is shown that ANN in general performs better than ARMA, MLP models perform better than recurrent, and the best training algorithm is found to be the cascade correlation. In Fonte *et al.* (2005) MLP are explored for very short term forecasting, and their suitability is once again confirmed.

Lahoz and San Miguel (2006) present one of the very few studies that present also wind direction as a forecast input and output. However, the average error is high ($\approx 50°$), suggesting that wind speed and direction might be less correlated than expected, at least in the medium term (6 hours).

Successive studies investigate alternatives to the MLP with BP training algorithm, for instance radial basis function networks (Chen *et al.*, 2009), shown to halve the error of MLP forecasting hourly averages, or Elman recurrent networks (Li *et al.*, 2010), for which the error is slightly lower than MLP, but it is shown this difference depends on the dataset used. Besides forecasting performance, Welch *et al.* (2009) point out that MLP have the non trivial advantage of being faster than more refined ANN models.

To improve ANN performance, some authors have combined ANN with other approaches. These include fuzzy logic (Monfared *et al.*, 2009), which also makes the model computationally more efficient, and Markov Chain models (Pourmousavi Kani and Ardehali, 2011). The latter is the study presenting the shortest term forecast, with time steps of 2.5 s. Some authors show how suitable data pre-processing can improve the forecasting ability of the ANN. For instance this is done in Fesharaki *et al.* (2010) by using weighed particle swarm optimisation to increase the amount of data to feed to the ANN, in Akinci (2011) by training a different MLP for each month of the year and forecast daily average speeds, and in Guo *et al.* (2012) by obtaining sub-time series from the original data set using empirical mode decomposition.

Other studies focus on getting the most out of available data which may include measurements form other locations in applications that require very short, short and long term forecasts (Barbounis *et al.* (2006); Barbounis and Theocharis (2007); Bilgili *et al.* (2007)).

A huge number of other case studies that use data recorded from all over the world are available. There is a general agreement in the better forecasting ability of ANN models compared to persistence and ARMA models for time ranges spanning from very short term (2.5 s) to long term (monthly averages). The MLP is the most explored structure, and although some studies show better performance from recurrent or Elman structures, in the majority of cases this leads to a slower model.

Unfortunately, there is no standard error measurement, and it is difficult to compare results by different authors. There are no general rules for choosing the optimal ANN structures, and because of the fragmentariness of those studies it is hard to deduct guidelines for building an efficient ANN mode for a new data set. In this work, a MLP structure is chosen, and all the parameters of the model will be optimised.

2.3.2 Support Vector Machines

Support vector machines (SVM) constitute a class of supervised learning models similar to ANN. SVM were originally developed for solving classification problems, where data need to be classified in two or more different categories, and have successively been used also to tackle regression problems (Muller *et al.*, 1997). Regression using SVM is also called in the literature support vector regression (SVR).
SVM are based on a map of the data into a higher dimensional space where the problem is linearly separable. The problem is then solved in the new space. In the case of binary classification this process can be visualised as in Fig. 2.3. In this simple example, points on a plane belonging to two different categories are mapped into a three-dimensional space. The problem is not linearly separable on a plane, but it is linearly separable in three dimensions.

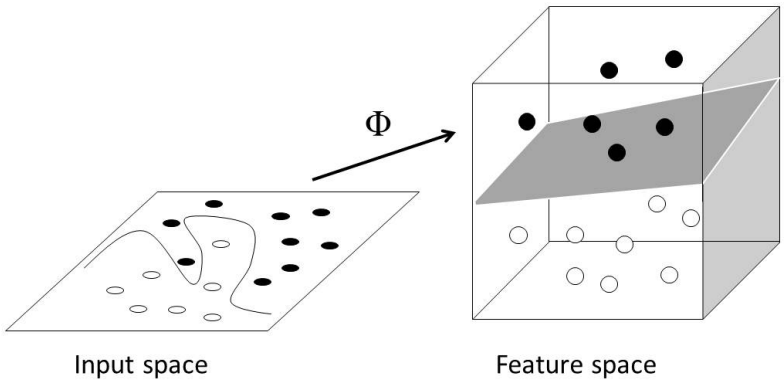

Figure 2.3: Mapping of 2D non linearly separable data to a 3D space where the problem becomes linearly separable - appeared in Tagliaferri *et al.* (2015).

This concept is generalised for SVR, where the aim is to model a function $y = f(\bar{x})$.
The use of SVR in this work is limited to a comparison with ANN, and there is no claim of novelty in the use of SVR as a forecasting technique. SVR has been successfully implemented in open-source software that makes possible also for non-expert user the use of this technique. An exhaustive description of SVM is out of the scope of this work, and in the reminder of this Section the basic equations defining the SVR process are presented. References are provided for the interested readers.

In this case we have to look for a function Φ which maps the problem to a higher dimensional space where it is possible to perform a linear regression, as shown in Equation (2.12):

$$y = \hat{f}(\bar{x}) = a \cdot \Phi(\bar{x}) + b \text{ where } \Phi : \mathbb{R}^n \to \mathscr{F} \subseteq \mathbb{R}^{n+m},\ a, b \in \mathscr{F} \tag{2.12}$$

A solution to this problem is defined as a minimum for the error function shown in Equation (2.13):

$$R[f] = \sum_{i=1}^{n} C(f(\bar{x}_i) - \hat{f}(\bar{x}_i)) + \lambda |b| \tag{2.13}$$

where n is the sample size, C is a cost function and the term $\lambda |b|$ is added to enforce flatness in the higher-dimension space. The function Φ can be found as the unique solution of a quadratic programming problem (Muller *et al.*, 1997; Vapnik, 2000). Equation (2.12) can be rewritten in terms of this solution as

$$\hat{f}(\bar{x}) = \sum_{i=1}^{n} (\alpha_i - \alpha_i^*) k(\bar{x}_i, \bar{x}) + b \tag{2.14}$$

where k is a symmetric kernel function (see Schölkopf *et al.* (1997) for details) and α_i, α_i^* constitute the solution to a quadratic programming problem, which depends on the cost function C from Equation 2.12. A possible formulation for C is Vapnik's ε-insentitive loss function, which assumes the form

$$C(f(\bar{x}) - y) = \begin{cases} |f(\bar{x}) - y| - \varepsilon & \text{for } |f(\bar{x}) - y| \geq \varepsilon \\ 0 & \text{otherwise} \end{cases} \tag{2.15}$$

The respective quadratic programming problem is defined as

$$\text{Minimise } \frac{1}{2} \sum_{i,j=1}^{l} (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) k(\bar{x}_i, \bar{x}_j) - \sum_{i=1}^{l} \alpha_i^* (y_i - \varepsilon) - \alpha_i (y_i + \varepsilon) \tag{2.16}$$

subject to $\sum_{i=1}^{l} \alpha_i - \alpha_i^* = 0$, $\alpha_i, \alpha_i^* \in [0, \frac{1}{l}]$. Details of this method can be found in Muller *et al.* (1997).

An important difference between SVR and ANN approaches is that SVR lead to a unique deterministic model for each data set, while ANNs depend on a random initial choice of synaptic weights. Therefore, in order to minimise the effects of this randomness in an ANN model, we train different ANNs and then average their outputs. This is not necessary for SVR, which constitutes an advantage in terms of computational time.

Chapter 3

Background on velocity prediction and aerodynamic interaction between two boats

In this Chapter the main challenges in single and multiple yacht modelling are presented and their relevance to the problem of racing strategy optimisation is discussed. The first Section of this Chapter introduces single yacht modelling and velocity prediction. The second Section focuses on multiple yachts, describing models for the physical interactions between yachts but also presenting the methodology of *risk coherent measures*, that are used in this work to include risk modelling in the racing strategy optimisation.

3.1 Velocity prediction

Throughout this work the knowledge of the boat speed for any given wind speed and direction is assumed. This is considered equivalent to "knowing the polars of the boat", where the term "polars" refers to a set of curves of the kind shown in Fig. 1.1. The yacht's polars can be obtained as the output of a Velocity Prediction Program (VPP). A VPP is a computer program for computing the maximum speed of a sailing yacht in any given wind conditions, together with a set of other important quantities such as, for instance, the heel and leeway angles at which the maximum speed is achieved. VPPs have a variety of applications, as they are useful to yacht designers to evaluate and compare different candidate designs, they are used to determine handicapping rules and they can help sailors to get guidance on optimal boat and sail tuning.

Figure 3.1 shows a yacht (as seen from above), with the forces acting on it on the horizontal plane. The motion of the yacht in the water generates the hydrodynamic force that can be decomposed in resistance, aligned with the BS but acting in the opposite direction, and hydrodynamic side force, perpendicular to the resistance. Similarly, the interaction between the wind and the sails generates the aerodynamic force, whose components are a drag, parallel to the apparent wind direction (i.e. the wind experienced by the yacht and resulting from the vector difference between the true wind velocity and the boat velocity), and a lift, perpendicular to the drag on the horizontal plane. The total hydrodynamic and aerodynamic forces are shown in

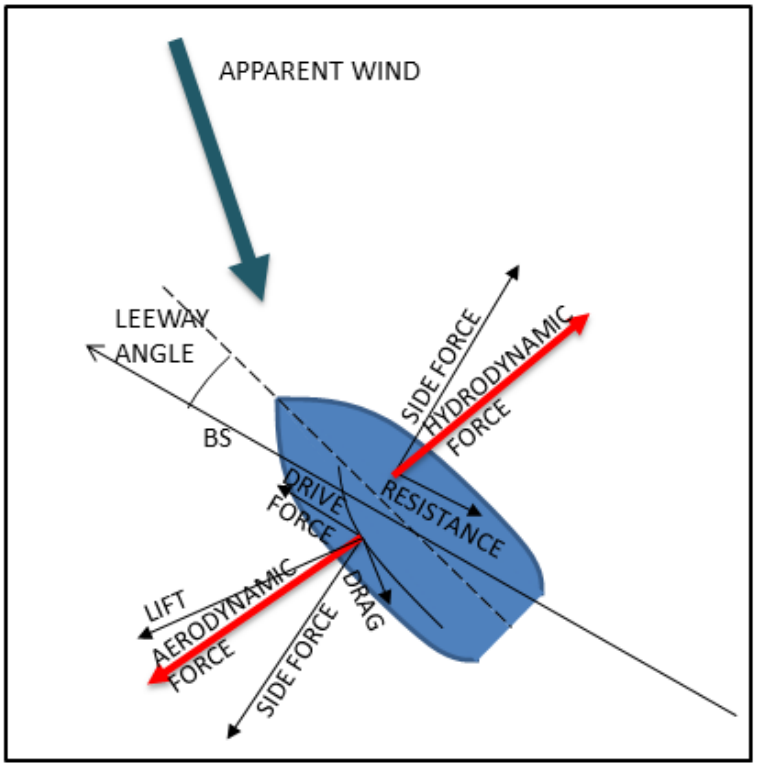

Figure 3.1: Forces acting on a sailing yacht.

red. In steady conditions, forces acting in opposite direction must be balanced. Therefore, the projection of the aerodynamic force on the direction of motion (drive force) must balance the resistance, and the hydrodynamic side force must balance the projection of the aerodynamic force on the opposite direction.

Figure 3.2 shows the action of the moment generated by the aerodynamic and hydrodynamic side forces, which have the effect of heeling the boat (heeling moment). This is contrasted by the moment due to the action of gravity and buoyancy (righting moment). A sailing yacht can be therefore seen as a dynamic system with six degrees of freedom, describing the motion in the the three directions and the rotation around the three axes (not shown in the Figure). Most VPP, including the one presented in this Thesis, focus on three degrees of freedom: the forces equilibrium on the horizontal plane, and the equilibrium of rotation shown in Figure 3.2.

The forces acting on a sailing yacht have been known since the first half of the past century, and are thoroughly described in the literature. In-depth descriptions can be found, for instance, in Marchaj (1979) and Marchaj *et al.* (1982).

A sail can be modeled as a highly cambered wing with no thickness, immersed in a flow disturbed by the presence of the mast. Figure 3.3, from Viola and Flay (2011), shows a schematic representation of the pressures on a jib, mainsail, and asymmetric spinnaker. The pressures are

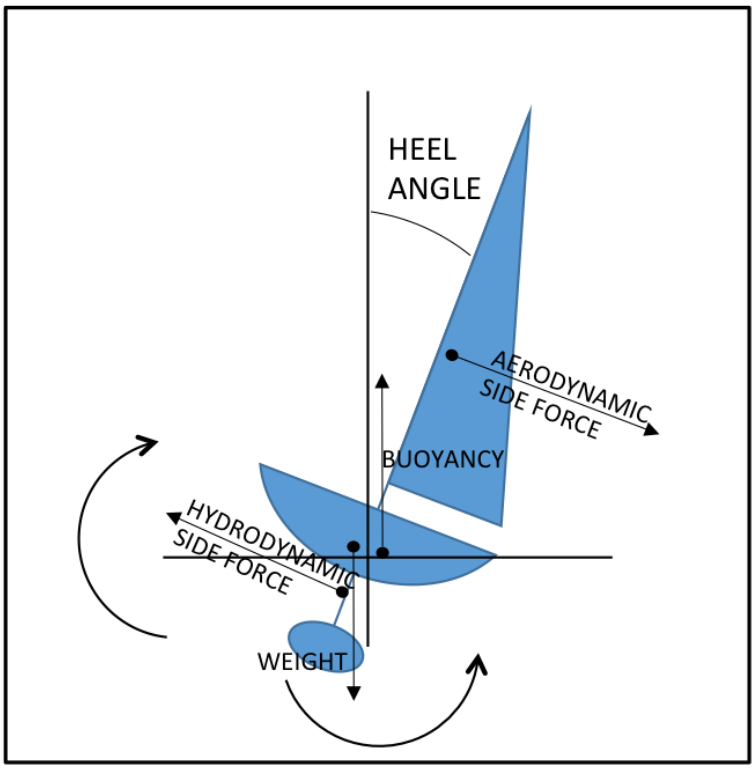

Figure 3.2: Momentum equilibrium

expressed in terms of a non-dimensional pressure coefficient, defined as:

$$C_p = \frac{p - p_\infty}{q_\infty} \tag{3.1}$$

where p is the static pressure, p_∞ is the free-stream static pressure, and q_∞ is the free-stream dynamic pressure. A detailed explanation of the physics of sails is out of the scope of this work, and can be found, for instance, in Larsson *et al.* (2014) and Viola (2013).

Numerical simulations and wind tunnel experiments are effective but expensive ways to compute aerodynamic forces. Similarly to hydrodynamic forces, regression based on experiments carried out on a set of different sails shapes can be used. The model developed by Hazen (1980) and used, among others, by Larsson *et al.* (2014), by ORC for the IMS handicapping rules, and in this thesis, allows the approximation of the aerodynamic forces as functions of apparent wind. Details are given in Chapter 6.

Computational Fluid Dynamics (CFD) and towing tank test can be used to determine the hydrodynamic forces acting on a hull, but these techniques are expensive in terms of cost and time. A fundamental contribution for the computation of the hydrodynamic forces acting on a yacht is given by the experiments carried out in Delft University of Technology in the Netherlands. Professor J Gerritsma and colleagues developed in the 1970s a series of 22 yacht models with systematic variations of five different hull parameters, performing towing tank

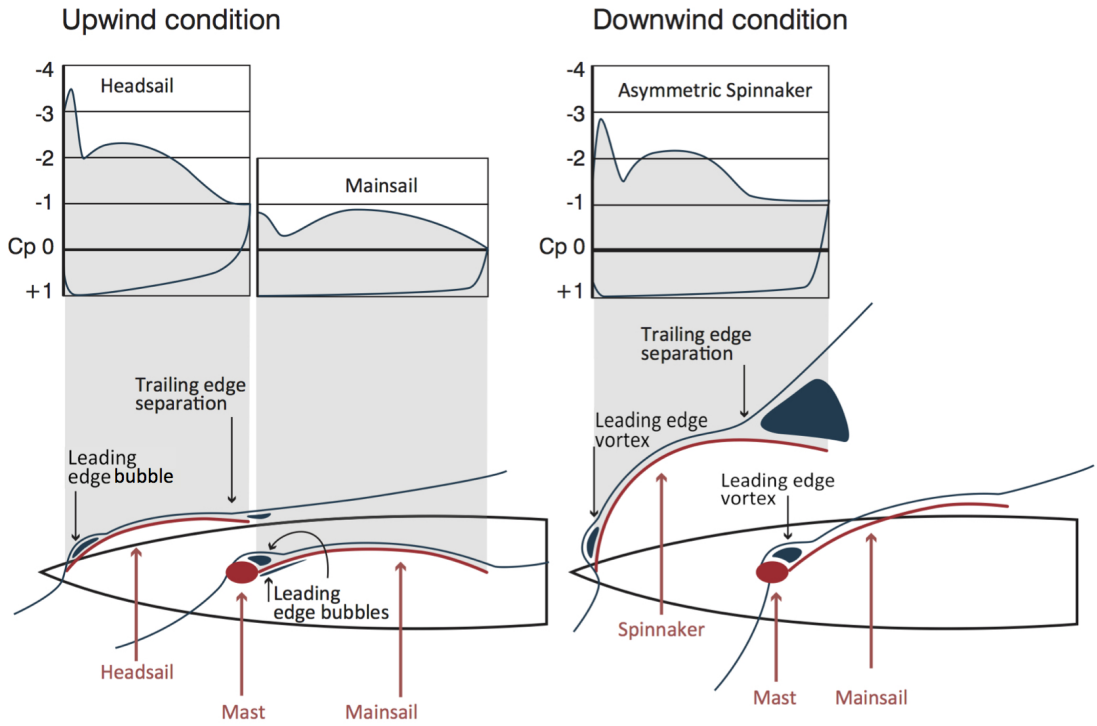

Figure 3.3: Schematic representation of flow and pressures around the sails (Viola and Flay, 2011).

tests to obtain regression coefficients for the computation of the resistance. The series was extended to 39 models in the 1980s and subsequently to 50 hulls, including also different keel and rudder models. The results of the experimental work are presented in Gerritsma *et al.* (1981); Gerritsma and Keuning (1985); Gerritsma *et al.* (1993); Keuning *et al.* (1996); Keuning and Binkhorst (1997); Keuning and Sonnenberg (1998). These data allow an estimate of the hydrodynamic forces based on the geometry of the boat, leeway and heel angle, and boat geometry. The simplicity and relatively high accuracy of this empirical formulations made them very popular in yacht design and have been used as the aforementioned aerodynamic model, among others, by Larsson *et al.* (2014), by ORC for the IMS handicapping rules, and in this thesis.

3.2 Aerodynamic interaction between two boats

A major concern for a skipper during a yacht race is its position with respect to the opponent when the yachts are very close. In fact, the presence of a second yacht causes a disturbance in the wind that may affect the flow experienced by the sails. This is often addressed in sailing-related literature (Whidden, 1990), and sailors are familiar with diagrams such as the one presented in Johnson (1995) and reproduced in Figure 3.4. There are two typical effects that the presence of a yacht can have. The first one is the significantly lower dynamic pressure in

the area downwind of the influencing yacht. The second one is a significant change in wind direction. Sailors refer to these phenomena as blanketing (represented by the black area in the left example of Figure 3.4) and backwinding (grey area), respectively. These effects usually leads to the situation where the second yacht must slow down and/or change slightly its course, dropping behind the leeward yacht. The influence decays with the distance from the influencing yacht but extends over more than ten boat lengths.

Both experimental and numerical approaches have been used to model the aerodynamic inter-

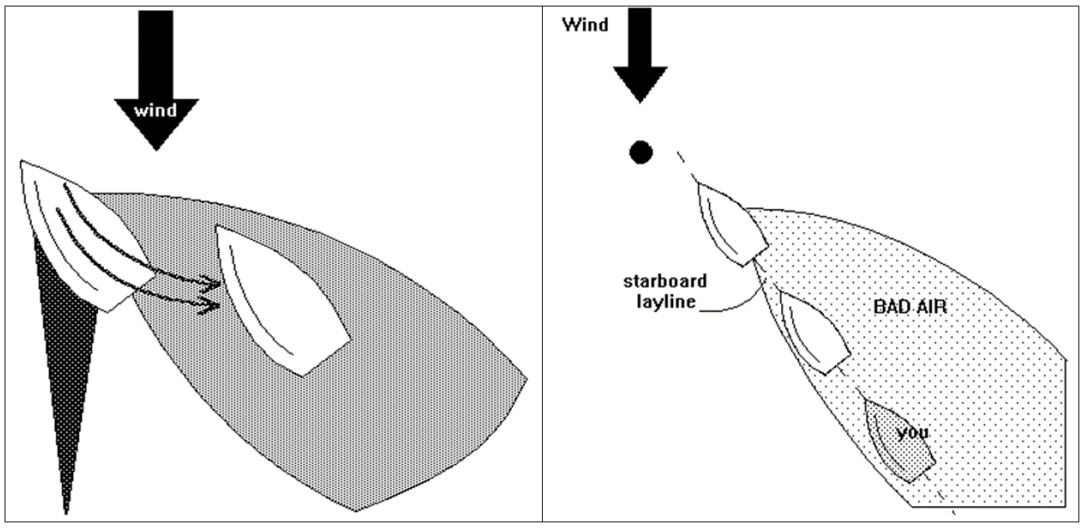

Figure 3.4: Interference zones as described by Johnson (1995).

action between boats, although only a limited set of situations have been tested.

An early example is in the book by Marchaj *et al.* (1982), where situations similar to the ones in Figure 3.4 are discussed. In this description, "safe leeward" and "hopeless" position are identified, but it is noted that also two areas of positive influence can be found, where the modified flow leads to an increase of boat speed for the windward yacht. Experimental wind tunnel results were performed at AWA of 40°, which are typically reached for a TWA of 90° (when reaching). Those results confirmed that the wake of the boat extends along the apparent wind direction rather than the true wind direction as in Johnson (1995).

The numerical approach is used in Caponnetto (1995) to study IACC yachts sailing at an AWA of 25° (upwind). The influencing yacht is fixed, and the influenced boat is positioned at distances of one or two boat lengths for all angles. Results confirm that the region of influence is centered almost around the apparent wind line. Again, a region of positive influence is found, where the influenced yacht has a gain of 4% in drive force. Interestingly, a position in which both yachts get the same reduction in speed is found when their relative angle is 96°. It is pointed out that such situation allows both yachts to sail at the same speed, which is however 3% lower than the speed of the isolated yachts (therefore, in a fleet race, both yachts would be disadvantaged).

Parolini and Quarteroni (2005) also use a numerical approach to model IACC yachts, and in particular they focus on the visualisation of the perturbed flow in downwind sailing. The two trailing vortices generated at the head and base of spinnaker and mainsail appear to have the greatest influence on the wind disturbance.

Philpott *et al.* (2004) base their model on the description of Twiname (1983), and in particular on the *bent air effect*. The wind speed is assumed not to be affected, while the wind direction is affected in an area downwind with respect to the influencing yacht of radius R_{bent} according to the following Equation:

$$TWD_{bent} = TWD + (CWA - \delta)\left(1 - \frac{r}{R_{bent}}\right) \tag{3.2}$$

where TWD and TWD_{bent} are the true wind directions as seen from the influencing and the influenced yacht, respectively, δ is the boom angle, and TWA is the course wind angle of the influencing yacht. An example of bent air effect is shown in Figure 3.5.

Spenkuch *et al.* (2011) develop a model, based on lifting line method, for the interaction

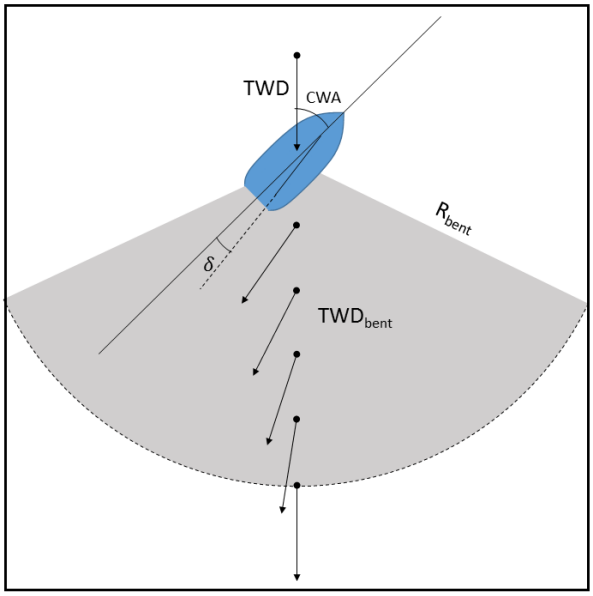

Figure 3.5: Bent air effect from Eq. 3.2

between multiple yachts, to be implemented in the *Robo-Race* simulator (Scarponi *et al.*, 2007a). The aim of this work is to give a detailed description of the wake of a sailing yacht, in order to include the modified flow's characteristic in the computation of the BS. In this study, "the wake of an upwind sailing yacht is represented as a single heeled horseshoe vortex and image system. At each time step, changes in vortex strength are convected into the wake as a pair of vortex line elements; these subsequently move in accordance with the local wind, the self-induced velocity and the velocity induced by the presence of the wakes of other yachts."

Richards *et al.* (2013) very recently studied the interaction between yachts via a set of wind

tunnel experiments, carried out in the Twisted Flow Wind Tunnel of the University of Auckland. Wind angles and boat distances tested are limited by the geometry of the wind tunnel, and both upwind and downwind cases are investigated, with AWA of 20° and 60°, respectively. Results for the upwind case are compared with the aforementioned studies, confirming the trends found by Caponnetto (1995) but with slightly higher variations in driving force. As also noted by Marchaj *et al.* (1982), the centre of the area of negative influence is a few degrees off the AWA line, implying that the wake has been slightly deflected by the lift generated on the sails (to be noted that in wind tunnel tests the boat is fixed and the boat heading is set at the AWA with the wind tunnel longitudinal axis). Moreover, an area of positive influence is found in the lee bow position. In the downwind case, this study shows that the region of influence lies across the true wind rather than along it. A schematic representation of the model arising from these results and presented in Richards *et al.* (2013) is shown in Figure 3.6.
The results from Richards *et al.* (2013) which have been implemented in the present study allow the definition of areas of positive influence and an indication on the speed reduction. The model used in this work for the physical interaction is described in Section 6.2 in the methodological part of this thesis.

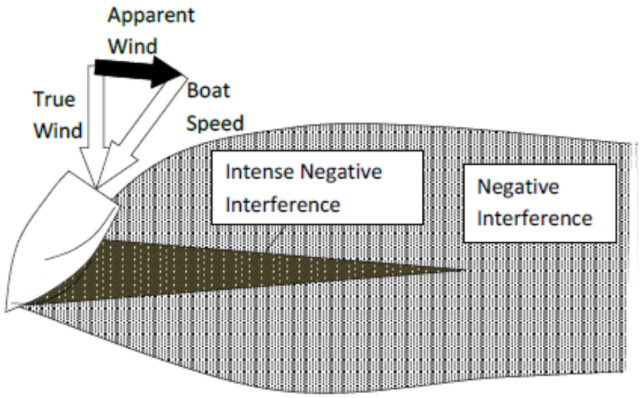

Figure 3.6: Re-elaboration of Figure 3.4 presented in Richards *et al.* (2013)

Chapter 4

Background on routing algorithms

4.1 An introduction to Dynamic Programming

The problem of taking optimal decisions during a yacht race can be seen as a multi-stage decision process. Problems belonging to this class have been extensively studied since the 1950s, and a very successful approach has been Dynamic Programming (DP), introduced by Bellman (1952) and developed in the following years. DP is a methodology to find solutions to a sequential problem that satisfy the following *Principle of Optimality: An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision.*
In this Chapter, the basics concepts to understand DP are given. The first Section describes the the functional equations and a basic computational scheme. The descriptions and equations in Section 4.1 very briefly summarise the work published by Bellman (Bellman, 1952; Bellman and Dreyfus, 1962). The previously explained concepts are then put into practice with an easy example. Section 4.5 summarises the previous work by Philpott *et al.* (2004) on the application of DP in racing strategy.
DP was developed to tackle a generic class of problems where a series of operations must be performed in order to achieve a desired result. Each one of these operations has a certain consequence, which can be stochastic. Typically, problems falling into this class aim at maximising a certain gain in a given time, or minimising time or costs needed to complete a task.
Let us consider a dynamic system evolving according to the following Equation 4.1:

$$x_{k+1} = f_k(x_k, u_k, \omega_k), t = 1, \ldots, N-1 \tag{4.1}$$

where k represents a discrete step, x_k and x_{k+1} represent the state of the system at steps k and $k+1$, respectively, u_k represents a decision, also called *control*, and ω_k is a random variable influencing the evolution of the system, characterised by a certain probability function p_k. The step index may refer to an increment over time or space, and the increment doesn't need to have fixed amplitude. Usually the initial state x_0 is fixed. All the variables defined take values in some determined interval or space; in particular, for a given state of the system x_k, the set of admissible controls $\mathscr{U}_k(x_k)$ is defined as the set containing all the possible decisions that can be taken at that stage. For instance, in financial problems, $\mathscr{U}_k(x_k)$ may be the set of all the possible assets that it is possible to buy or sell. In sailing applications, x_k can represent the state of a

yacht on the race area (in this case x_k can be the vector constituted by the yacht's coordinates and the observed wind, assuming values on a limited subset of $\mathbb{R}^n$), u_k the CWA followed by the skipper ($u_k \in \mathscr{U}_k \subseteq [0,360)$), and ω_k the unknown wind evolution between step k and step $k+1$. The position of the yacht at step $k+1$ is then a function of those three variables.
A control, or a *policy*, is a finite sequence $U = u_0, \cdots, u_{N-1}$, where $u_k = u_k(x_k)$ is a function of the current state of the system, and all the $u_k \in \mathscr{U}_k(x_k)$ for all x_k. In the following, $\mathscr{U}$ will denote the set of the admissible policies.
The aim of DP is to find an admissible policy $U = u_0, \cdots, u_{N-1}$ that minimises a cost function which can assume the generic form as expressed in Equation 4.2:

$$C(U,\omega) = \sum_{k=0}^{N} c_k(x_k, u_k(x_k), \omega_k) \tag{4.2}$$

where $\omega = [\omega_0, \ldots \omega_N]$, subject to the system constraint specified in Equation 4.1. In sailing, this cost corresponds to time:

$$T(U,\omega) = \sum_{k=0}^{N} t_k(x_k, u_k(x_k), \omega_k) \tag{4.3}$$

where t_k represents the time needed to sail from state x_k to state x_{k+1}.

4.2 An example of DP execution

This section is dedicated to the description of the principle of optimality via a simple and illuminating example, constituted by a shortest path problem. This example is aimed at helping the reader that is not familiar with the DP algorithm, and can be skipped without loss of continuity. Let us consider the graph in Figure 4.1. In this deterministic case, the aim is to get from node A to node N in the shortest possible time, where the time needed to get from one node to another is shown on the corresponding edge. As the costs are deterministic, all the expected values can be omitted. Due to the geometry of the graph, this problem is a six-stages problem. With the notation used in the previous Section, at step 0 the system is in state x_0 =A, at step 1 x_1 =B or C, and so on until x_6 =N. We will now proceed to the computation of the optimal policy, starting from the final stage and proceeding backwards. The last stage is trivial, because nodes L and M have direct access to the final node N.

$$\begin{aligned} \mathscr{U}_5(L) = \{u_5(L) = N\},\, u_5^{opt}(L) = N,\, c_5(L, u_5^{opt}(L)) = 4 \\ \mathscr{U}_5(M) = \{u_5(M) = N\},\, u_5^{opt}(M) = N,\, c_5(M, u_5^{opt}(L)) = 6 \end{aligned} \tag{4.4}$$

Let's now proceed to the previous stage, where the possible states are I, J and K. The optimal action for nodes I and K is again trivial because there is only one possible choice (L and M,

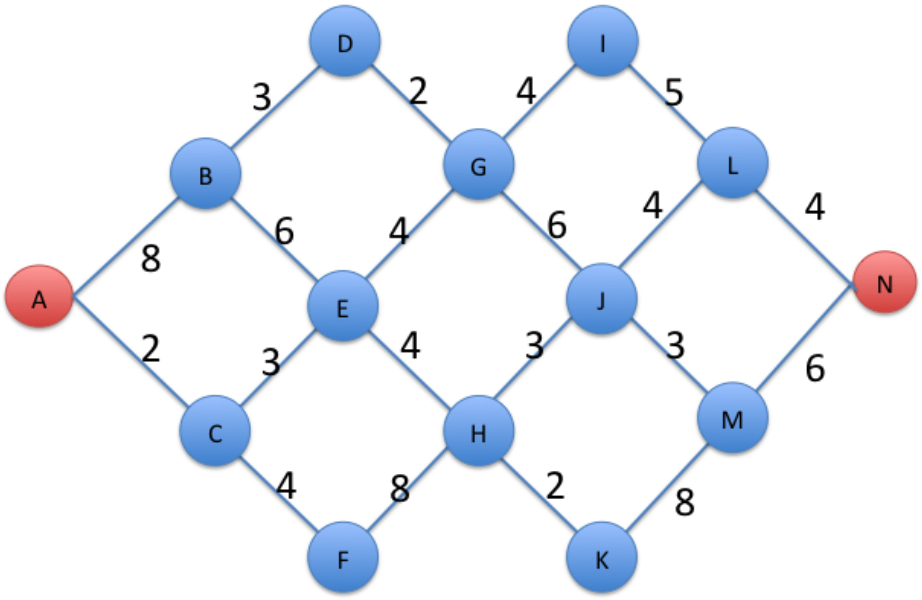

Figure 4.1: Example of shortest-path problem.

respectively). For node J there are two possible choices:

$$\mathscr{U}_4(J) = \{u_4^1(J) = L, u_4^2(J) = M\} \tag{4.5}$$

Recalling the principle of optimality, the optimal decision made at stage 4 must be optimal for the problem starting at that stage. Therefore

$$\begin{aligned} C(u_4^1(J)) &= c_4(u_4^1(J)) + c_5(u_5^{opt}(u_4^1(J))) = 4+4 = 8 \text{ (through node L)} \\ C(u_4^2(J)) &= c_4(u_4^2(J)) + c_5(u_5^{opt}(u_4^2(J))) = 3+6 = 9 \text{ (through node M)} \end{aligned} \tag{4.6}$$

and the optimal action for node J is then

$$u_4^{opt}(J) = L \tag{4.7}$$

because it is the action that minimises the cost for the problem starting from node J. The procedure is now iterated until the node A. An easy way of visualising this process is by assigning to each node its optimal (minimum) cost, i.e. the minimum time needed to get from that node to the final node N. The result is shown in Figure 4.2. The optimal policy is therefore $U = \{A, C, E, H, J, L, N\}$.

This example underlines some important features of DP. First of all, despite the seemingly difficult recursive structure the procedure is straightforward. Taking again node J as an example,

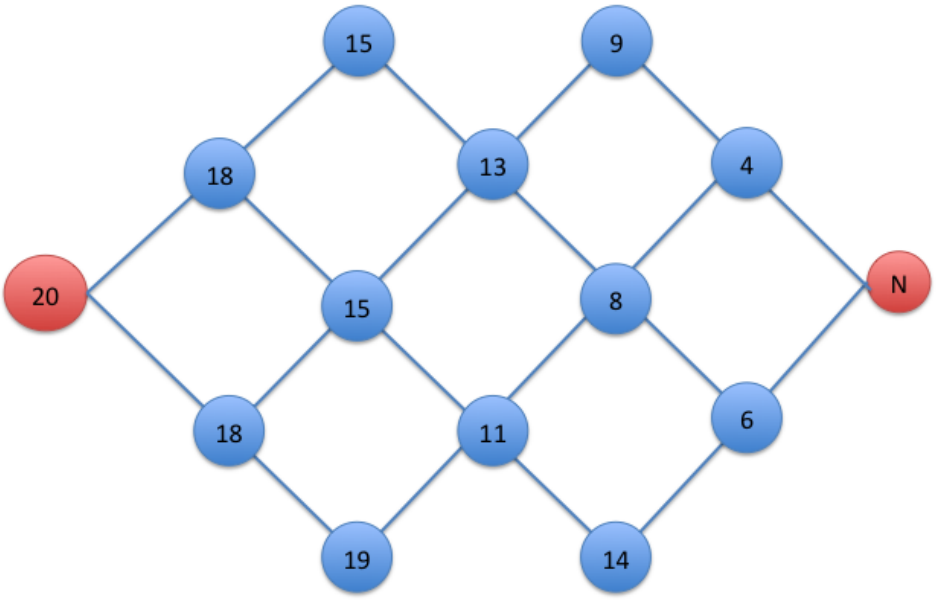

Figure 4.2: Example of shortest-path problem.

the optimal policy includes a step which is apparently suboptimal if the cost associated to the step is the only quantity considered. In fact, going from node J to node L is more expensive than going to node M. However, DP considers the cost-to-go, obtaining the optimal choice. This means that DP is able to model a common situation in sailing races, when a skipper decides to temporarily sail in an unfavorable wind, to be able to tack into a more favourable wind that can lead to an advantage (a situation of this kind was described in Figure 1.3). Moreover, DP is significantly more efficient compared to an exhaustive search, which would require the computation of all possible paths from the initial to the final node and the associated cost. A deeper insight on the computational cost of the general DP algorithm can be found, for instance, in Bellman and Dreyfus (1962).

4.3 Stochastic dynamic programming

The previous example belongs to the class of deterministic problems. For this class of problems, the cost function is known at every stage. Unfortunately, in practical applications (including sailing) the cost function is only known in terms of a probability distribution, and rather than minimising a cost the aim is to minimise its expected value. In this case, the stochastic version of dynamic programming is used. Going back to the general description, a solution for the

problem is then a policy U^{opt} such that

$$\mathbb{E}(C(U^{opt})) = \min_{U \in \mathscr{U}} \mathbb{E}(C(U, \omega)) \tag{4.8}$$

We assume that the minimum in Equation 4.8 is well defined. A discussion of this aspect can be found in Bertsekas *et al.* (1995). According to the principle of optimality, an optimal solution has the property that, considering the subproblem starting at stage M, then the sub-policy $(U^{opt,M} = (u_M^{opt}, u_{M+1}^{opt}, \ldots, u_N^{opt})$ is optimal for that subproblem. This principle has its justification in the following derivation.

$$\begin{aligned}
\min_{U \in \mathscr{U}} \mathbb{E}(C(U, \omega)) &= \min_{u_0,\ldots,u_N} \mathbb{E} \sum_{k=0}^{N} c_k(x_k, u_k(x_k), \omega_t) \\
&= \min_{u_N} [\min_{u_0,\ldots,u_{N-1}} \mathbb{E} \sum_{k=0}^{N} c_k(x_k, u_k(x_k), \omega_t)] \\
&= \min_{u_N} [\min_{u_0,\ldots,u_{N-1}} \mathbb{E} c_N(x_N, u_N(x_N) + \mathbb{E} \sum_{k=0}^{N-1} c_k(x_k, u_k(x_k), \omega_t)] \\
&= \min_{u_N} [\mathbb{E} c_N(x_N, u_N(x_N) + \min_{u_0,\ldots,u_{N-1}} \mathbb{E} \sum_{k=0}^{N-1} c_k(x_k, u_k(x_k), \omega_t)] \\
&= \min_{u_N} [\mathbb{E} c_N(x_N, u_N(x_N)] + \min_{u_0,\ldots,u_{N-1}} [\mathbb{E} \sum_{k=0}^{N-1} c_k(x_k, u_k(x_k), \omega_t)]
\end{aligned} \tag{4.9}$$

This derivation leads to defining a recursive algorithm for finding an optimal policy, proceeding backwards and solving subsequently truncated sub-problems. The process described in Equation 4.9 must not mislead the reader in thinking that the optimal solution can be found by minimising the sub-functions c_k. A locally optimal decision may in fact lead to a subsequent stage of the system from which only "bad decisions" can be taken. Although the expected values in Equation 4.9 are linear, and therefore the sums defining the function C can be decomposed, the interdependence between stages of the problems is hidden in the sets of possible decisions u_k.

4.4 Markov Chains

A key step in stochastic DP algorithms consists in the computation of the expected cost of a policy U. This requires the knowledge of the probability laws governing the evolution of the stochastic process ω. In practice, to know the expected time needed to sail from node i to node y we need to know the time needed for any possible wind realisation, and also the probability of each realisation, in order to compute a weighted average. The probability model used in this thesis is based on Markov Chains (MC), stochastic processes where the probability of an event depends on the realisation of the previous event.

Let $\omega_0, \omega_1, \ldots$ be the stochastic process representing the wind velocity at times $0, 1, \ldots$. The

Markov property states that

$$\mathbb{P}(\omega_k = w_k|\omega_{k-1} = w_{k-1}, \omega_{k-2} = w_{k-2}, \ldots, \omega_0 = w_0) = \mathbb{P}(\omega_k = w_k|\omega_{k-1} = w_{k-1}) \quad (4.10)$$

This property, also called *memory loss*, means that the probabilty distribution for ω at step k depends on the value assumed by ω_{k-1} but, importantly, not on the previous events.
As a consequence of Equation 4.10, a Markov Chain can be characterised in terms of a transition matrix, containing all the probabilities of the transitions between two states. Let $\mathscr{W} = w1, \ldots, w_n$ be the finite space of all the possible states for ω. For instance, if ω represents the wind direction only, the space $\mathscr{W}$ can be the set $\{1, 2, \ldots, 360\}$. For each couple of elements in $\mathscr{W}$ it is possible to define the probability p_{ij} of transitioning from state w_i to state w_j as follows:

$$p_{ij} = \mathbb{P}(\omega_k = w_j|\omega_{k-1} = w_i) \quad (4.11)$$

The matrix $P = \{p_{ij}\}$ is called the *transition matrix*. The initial state ω_0 cannot depend on any previous state, therefore an initial distribution is defined:

$$p_i^0 = \mathbb{P}(\omega_0 = w_i) \quad (4.12)$$

State space, initial distribution and transition matrix uniquely define a Markov Chain.

Figure 4.3 shows a common way of representing MC. The process *"jumps"* from one state to the next according to the probabilities associated to the arrows. It is clear from the representation that the transition probabilities depend on the current state, but not on the previous ones.
In this application MC can be used for a double aim, i.e. to statistically model wind and its properties in order to compute the expected values in the DP algorithm, but also to generate artificial wind time series to test the DP model. The first operation needed for using a MC model is to define a state space and then compute the transition matrix P and initial distribution p^0 from recorded data. The state space is usually defined by dividing the interval to which recorded data belong in some subintervals. Taking again the example of wind direction in Philpott and Mason (2001), with a methodology described also in Tagliaferri *et al.* (2014) and Tagliaferri (2015), the state space consists in bins having amplitude of 5°. Let us denote the bins by $w_1, \ldots, w_n$, coherently with the notation used above. Let $wd_1, \ldots, wd_n$ be a recorded time series of wind directions, where m is the total number of data points collected. Then the initial distribution is computed as the cumulative distribution of the total data set. Let d_i be the number of values falling in bin w_i. Then

$$p_i^0 = \frac{d_i}{n} \quad (4.13)$$

Similarly, the transition matrix P is computed by using a maximum likelihood estimator as follows: let n_{ij} be the number of occurrences of subsequent values falling in bins $w_i w_j$ and let

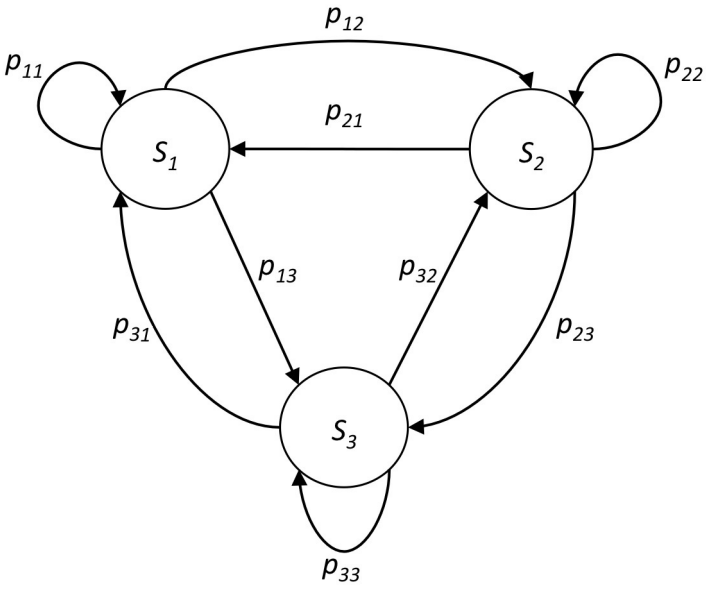

Figure 4.3: Representation of a Markov Chain with three states

d_i be the number of values falling in bin w_i. Then

$$p_{ij} = \frac{n_{ij}}{d_i} \tag{4.14}$$

As argued in Tagliaferri (2015) the bins don't have to be of the same amplitude, and their number and amplitude depends on the application.

4.5 Application of DP to yacht routing

Dynamic Programming was used to model sailing tactics by Prof Philpott and colleagues from the University of Auckland. This work is described in the publications by Philpott *et al.* (2004) and Philpott and Mason (2001). The DP algorithm is implemented in a computer program called SCORE (Short Course Optimal Routing Engine). As described in the methodology Section in Philpott and Mason (2001), the race area is discretised in a set of $N_x \times N_y$ nodes indexed by$(i,k)$where $i = 1,\ldots,N_x, n = 1,\ldots,N_y$, $N_x = 80$ and $N_y = 50$. Each node has coordinates $(x(i),y(k))$; as all the routes visit the nodes in order of increasing k, the index k refers to the stage of DP with the same notation as in the previous section. An additional state variable needed is $j \in \{-1,1\}$ that indicates whether the boat is sailing on a port or starboard tack. The wind speed is assumed to be constant, while the focus is on wind direction variation, modelled as a Markov Chain. At each stage the wind is constant across the course, but there is a random

transition (according to the Markov probability model) at every step.
The set of possible decisions at each step is the set $\mathscr{U}_k(i)$, which includes all the nodes (denoted generically as y) that can be reached from the node (i,k). Let $t_k(i,y,j)$ denote the time needed to sail from the node (i,k) to the node $(y,k+1)$ on tack j, and let τ denote the time loss associated to a tack. The deterministic version of the DP algorithm uses the following recursion:

$$f_k(i,j) = \min \begin{cases} \min_{y\in\mathscr{U}_k(i)}[t_k(i,y,j) + f_{k+1}(y,j)] \\ \min_{y\in\mathscr{U}_k(i)}[t_k(i,y,j) + f_{k+1}(y,-j)] + \tau \end{cases} \tag{4.15}$$

The function $f_k(i,j)$ represents the time-to-go, i.e. the minimum cost associated to each node as in the example of Figure 4.2. Equation 4.15 assumes knowledge of the time needed to sail between two nodes. However, this time is better modeled as a function of the stochastic wind. Again using the notation defined in the previous Sections, let ω_k denote the wind at time k. Then

$$f_k(i,j,\omega_k) = \min \begin{cases} \min_{y\in\mathscr{U}_k(i)}[t_k(i,y,j,\omega_k) + \mathbb{E}\{f_{k+1}(y,j,\omega_{k+1})|\omega_k\}] \\ \min_{y\in\mathscr{U}_k(i)}[t_k(i,y,j,\omega_k) + \mathbb{E}\{f_{k+1}(y,-j,\omega_{k+1})|\omega_k\}] + \tau \end{cases} \tag{4.16}$$

where the expected values are taken over all the possible values of ω_{k+1}, conditioned to the realisation of ω_k. The computation of this expected value is detailed in Section 4.4. The possibility of introducing some variations on the functions f_k to account for the risk attitude of the sailor is mentioned but not explored in detail. The authors cite the methodology presented by White (1987) to treat the risk by adding a new variable accounting for the time of arrival at each node.
The computer program implemented by the authors uses the recursion defined in Equation 4.16 to compute the optimal heading. Optimal policies are generated for a number of cases, showing the robustness of the method, and how the policies minimise the expected time to the mark.

4.6 Risk management

The objective of stochastic DP is to take a set of sequential decisions $U^{opt} = u_1, u_2, \ldots, u_N$ such that the expected value of the time T needed to complete the race, which depends on U but also on some random values (wind realisations), is minimised:

$$\mathbb{E}(T(U^{opt},\omega)) = min_{U\in\mathscr{U}}\mathbb{E}(T(U,\omega)) \tag{4.17}$$

where $\mathscr{U}$ represents the space of all possible decision sequences U. Given a wind realisation ω, the time is a deterministic function of ω and of the decisions sequence U. The expected time is computed as a conventional expected value, by multiplying the time computed for a realisation

ω_i and the probability of said realisation:

$$\mathbb{E}(T(U,\omega)) = \sum_{\omega_i \in \Omega} T(U,\omega_i)\mathbb{P}(\omega_i) \tag{4.18}$$

This corresponds to the objective of a sailor trying to complete a race in the shortest possible time *on average*. Aiming at minimising the average time means essentially invoking the Law of Large Numbers, and the fluctuations in the outcomes of the single races are negligible. Unfortunately, this is not really a convenient approach when dealing with competitions. This can be understood with a very simple example. Table 4.1 shows the arrival times of two yachts in a match race. The results shown indicate a set of very close races, won by A in all cases

Table 4.1: Example of arrival times

Race	1	2	3	4	5	6	7	8	Average
Boat A [min]	45.2	41.8	35.9	43.2	51.4	37.3	46.9	38.3	42.5
Boat B [min]	45.4	42.1	35.1	43.5	52.6	34.3	47.2	38.5	42.3

except 2 (race 3 and race 6). The average time needed to complete the race is higher for A, as shown in the last column of Table 4.1. It is an apparently counterintuitive result: even if A is slower on average, she wins most of the times. This suggests that minimising the expected time might be effective in a long set of fleet race, but it can be not the most practical option in a short set of match races.

This apparent contradiction is common in other applications, such as finance and operations research. Many example describing the computation of an optimal investment policy can be found in the literature, for instance in Shapiro and Ruszczynski (2008) . The results mostly suggest concentrating the investments in the assets with the highest expected return rate, which unfortunately are also the ones involving the highest risk of losing all invested money. In practice, when trying to generate a policy that maximises an expected gain, the computed solution can result in a probability of damage higher than the user is willing to accept, as argued in Philpott *et al.* (2013). The approach consisting in the minimisation of the expected value of a certain cost (i.e. time) is usually referred to as *risk neutral*.

A possible way to overcome this problem is to add some constraints to the problem, to avoid obtaining solutions that are not applicable in practice (Guigues and Sagastizábal, 2013). However, this approach is applicable only on small-scale problems.

Artzner *et al.* (1999) introduced another approach based on the use of *risk measures*, real-valued functions from the probability space underlying the problem. In this work, focused on the study of risk in financial market, the authors define a risk measure to be *coherent* if it satisfies four axioms. Without going into details, these axioms ensure that the convexity properties of the original optimization problem are preserved. This approach has recently been used together with dynamic programming to successfully solve multi-stage optimization problems (Philpott *et al.*, 2013). A key feature of this approach, which overcomes the need of an explicit definition for the risk measure, is that it is based on the dual representation theorem defined by Artzner *et al.* (1999). Thanks to this representation the risk measure computation is replaced by the use

of a modified probability distribution for the computation of the expected values.
The development of coherent risk measures was motivated mostly by applications in finance, where the aim is to compute optimal investment strategies that avoid high-risk decisions. As underlined in the previous Chapters, in a match race the aim of a skipper is to complete the race before the opponent. Sailors at different stages of a race might have different attitudes towards risk. For instance, consider the situation described in Figure 4.4, where the grey boat is leading with a significant time difference. She is close to reaching the layline, and she is soon going to tack and proceed on a starboard tack towards the mark. The black boat is left behind, and she is very likely to lose the race. A reasonable choice is to try and reach the left layline and tack towards the mark. However, let's assume that there is a positive probability, even if small, of a big wind shift towards the left. If this happens, and the black boat tacks just before, she will suddenly be in an advantaged position with respect to the grey boat, who is now in downwind from the mark. For the grey boat, the wind shift is "moving" the layline further away from its position, resulting in a disadvantage. As a general rule, if the yachts are sailing in the middle of the racing area the advantage (or disadvantage) caused by wind shifts will be less significant than if the yachts are sailing on opposite sides of the course.
Here the key point is that the time difference between the arrival times of the two boats is not

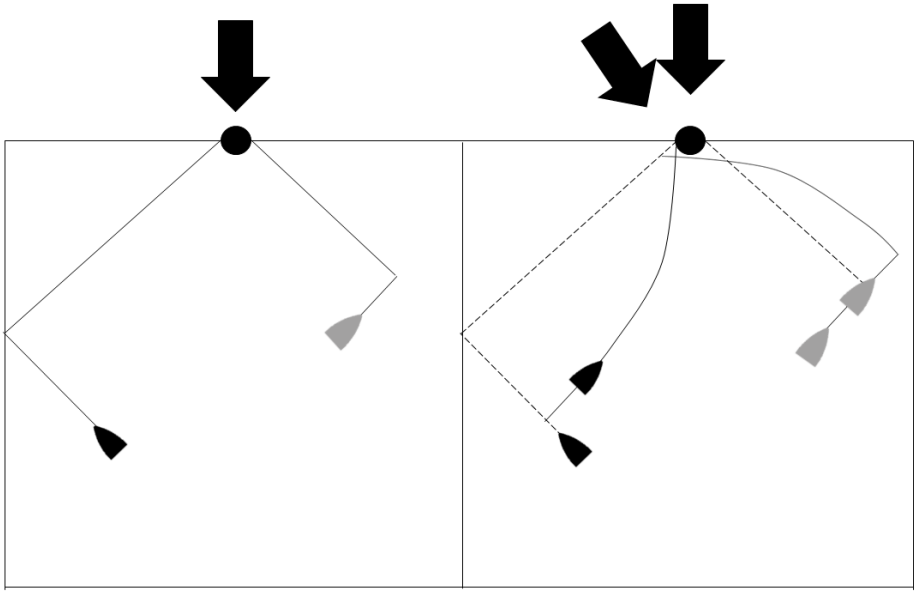

Figure 4.4: Example of risk-seeking attitude.

of interest, whereas its sign is. Therefore a skipper who is losing has no interest in trying and complete the race in the shortest possible time, as this is very likely to lead to a defeat anyway. The only chance for the black skipper is to be in the right position if the "hoped for" wind shift happens. Conversely, from the point of view of a winning boat, it is important to minimise the risk of losing the advantage gained so far. Therefore in this work two attitudes are investigated. A risk-seeking behaviour is associated to a skipper who is losing the race, while a risk-averse attitude is preferred by a skipper who is leading. The methodology for this investigation is

described in Chapter 8.

PART II

Method

Chapter 5

Wind forecasting

In this chapter the methodology used for wind forecasting is described. The wind velocity vector **w** is studied as a two-dimensional vector with components $ws(t)$ and $wd(t)$ representing wind speed and wind direction, respectively. The vertical component of the wind velocity is ignored because it has a minor effect on tactical decisions, although vertical wind variations must be taken into account for sail regulation.

5.1 Wind dataset

The wind dataset used in this work was recorded during the 34^{th} America's Cup races held in San Francisco in summer 2013. Data were collected during 23 races, corresponding to four Luis Vuitton's Cup semifinals between Italian team Luna Rossa Challenge and Swedish Team Artemis, five Luis Vuitton's finals between Luna Rossa and kiwi team Emirates Team New Zealand, and thirteen (one of which was repeated) races for the America's Cup finals. The dataset was publicly available on the website www.americascup.com in 2013, together with documents explaining data specification. The website is now focused on the 35^{th} America's Cup and the dataset is now offline and available upon request.

San Francisco was the chosen venue for the America's Cup events in summer 2013, which included the Louis Vuitton Cup, the competition that determines which team is going to have the opportunity to race against the current Cup defender. The races took place in San Francisco Bay, in an area delimited roughly by the Golden Gate Bridge, the southern shore, and Alcatraz Island. Figure 5.1 shows the area reserved to the event and the race area in the Bay (SF Coast Guard, 2013).

During summer, the wind direction in the Bay is, conveniently, almost constant. This means that the racing course doesn't change significantly over the days, allowing for better public organisation. Obviously, there are still wind direction variations during the race, constituting tactical challenges.

The dataset consists of 23 files, one for each race. Statistics for the dataset are summarised in Table 5.1. Data are logged at 5Hz frequency, and include on-board measurements of quantities such as boat speed and heading, and global measurements of wind speed and direction. According to the data specifications (ACEA, 2013) wind measurements are collected by the single boats, including racing yachts and committee boats, and on some of the marks, and then

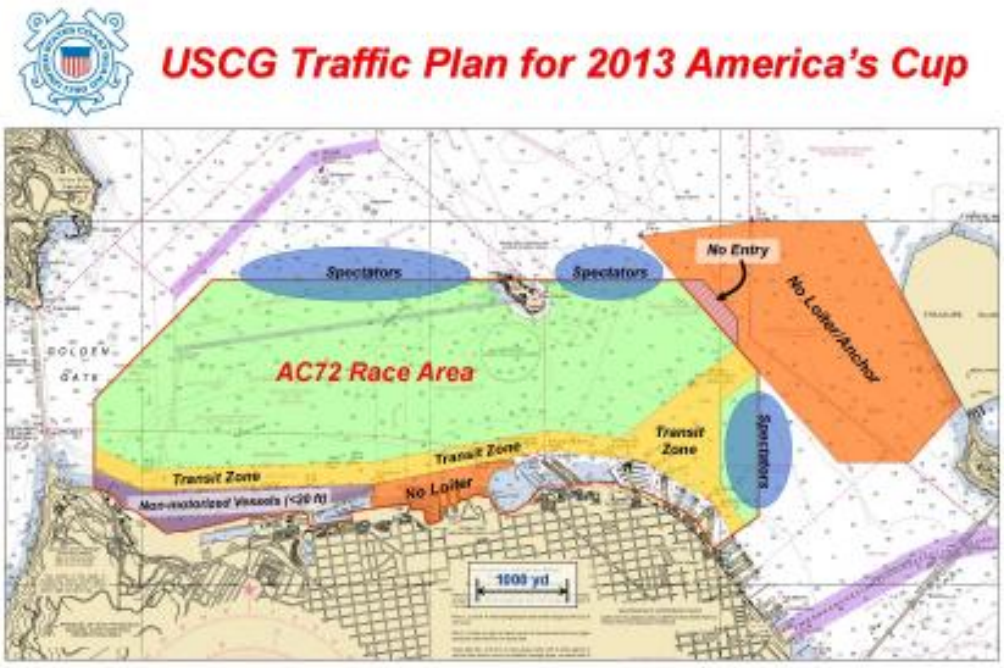

Figure 5.1: Racing area for the 2013 America's Cup events.

averaged over the course. The wind speed and direction obtained are the 'official' ones, used for graphics on TV broadcasting and by the umpires. The data files are ordered chronologically, therefore race 23 corresponds to the final race. The duration of the races varies, but most recordings are approximately one hour long, constituting a total of 21 hours of data.

Table 5.1: Wind data statistics

Race	Length	Wind speed [m/s]				Wind direction [deg]			
n	[min]	mean	std	min	max	mean	std	min	max
1	67.13	8.87	0.83	6.8	10.7	251.99	4.70	238.5	260.8
2	65.15	7.19	0.74	5.2	10.5	251.71	3.38	241.5	257.8
3	66.29	7.96	0.98	4.6	11.5	255.05	7.66	240.7	269.5
4	45.38	6.79	0.64	5.0	8.1	237.84	6.16	225.1	252.0
5	63.20	7.99	0.59	6.3	10.1	249.39	3.67	237.7	259.0
6	59.76	8.15	0.67	6.1	10.5	251.32	5.49	239.3	263.0
7	71.70	5.89	0.52	4.2	7.6	249.50	8.60	233.4	270.1
8	64.31	7.98	0.58	6.0	9.8	246.72	3.17	238.1	254.1
9	64.16	7.72	0.97	5.6	9.9	246.12	3.82	237.7	253.5
10	43.07	8.62	0.52	6.6	10.8	252.37	9.74	239.8	268.0
11	54.58	8.54	0.81	5.5	10.1	254.98	8.87	239.7	276.8
12	40.57	8.63	0.50	7.4	9.9	248.33	6.15	239.5	259.0
13	59.69	9.72	0.60	8.6	12.0	249.27	5.41	240.0	258.3
14	40.04	10.05	0.89	7.9	12.4	237.74	4.95	225.5	244.9
15	47.75	5.99	0.41	5.0	7.3	248.76	7.18	237.1	260.3
16	57.69	8.00	0.66	6.6	9.4	250.20	5.04	236.4	260.7
17	47.75	5.99	0.41	5.0	7.3	248.76	7.18	237.1	260.3
18	37.47	9.12	0.59	7.4	10.7	245.36	7.65	234.0	261.8
19	70.35	9.56	0.68	7.9	12.1	247.69	5.45	237.5	262.2
20	39.60	8.27	0.79	6.5	9.7	246.03	6.22	235.6	261.2
21	42.80	7.83	0.43	6.8	8.9	251.18	8.30	237.4	265.8
22	57.98	4.89	0.70	3.4	6.8	242.60	16.48	219.8	280.9
23	63.60	4.32	0.50	2.7	5.4	256.43	22.23	219.8	292.1

Wind speed varies between an overall minimum of 2.7 m/s and an overall maximum of 12.4 m/s. During the same race there can be wind variations of up to 7 m/s, although the standard deviation for the single data sets is never more than 1 m/s. The average wind direction is west-southwest (WSW). The variations in wind direction during one race can reach more than 60° (two cases) with a standard deviation of approximately 8°.

Figures 5.2 and 5.3 show an example of recorded wind direction and speed, respectively, corresponding to the final race of the AC events (race 23 in Table 5.1). The wind direction presents minor oscillations during the first part of the race, followed by a constant shift towards the right of approximately 50 degrees. Although this is an uncommon case for this dataset, this represents one of the typical examples of events that can affect the outcome of a race. The wind speed follows a different pattern, with a significant temporary lull corresponding to the beginning of the shift.

The correspondence of the speed lull with the beginning of the shift may suggest a correlation between those two events. The correlation between wind speed and wind direction was investigated in order to determine whether the use of both quantities as forecasting input may improve the forecasting ability of the model.

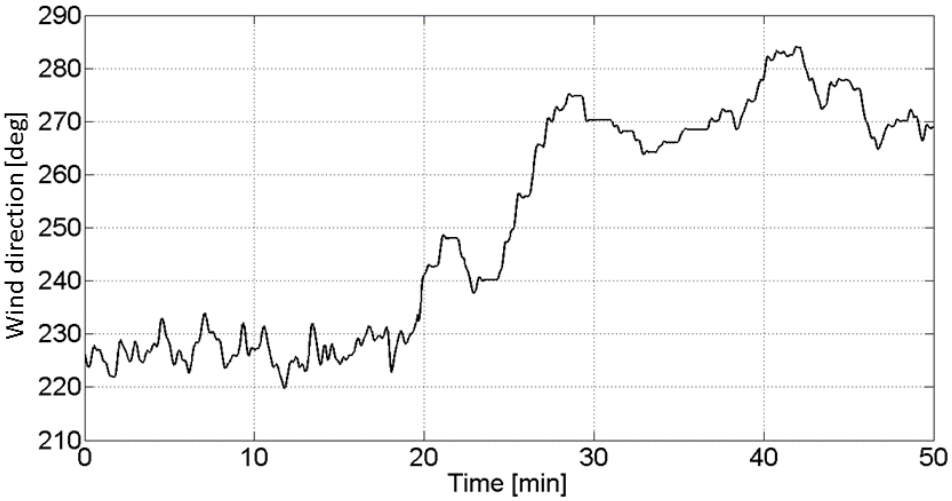

Figure 5.2: Example of wind direction dataset

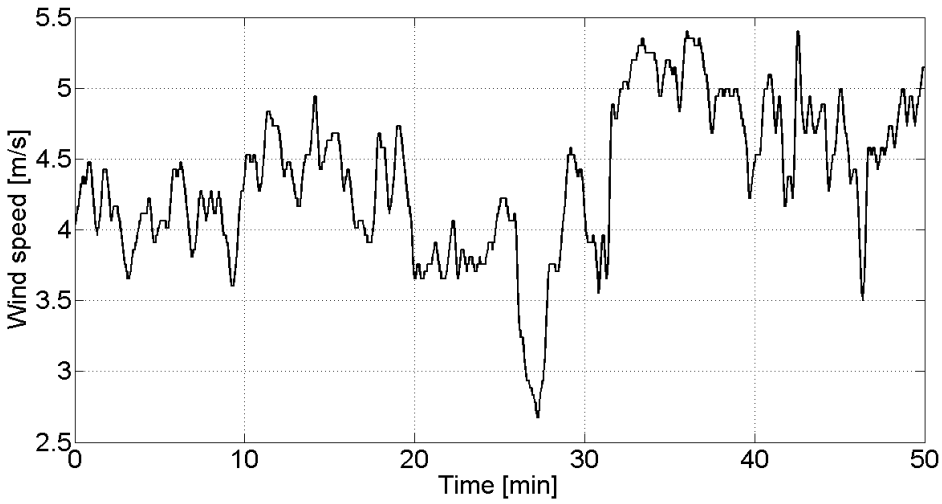

Figure 5.3: Example of wind speed dataset

Table 5.2 shows Pearson's Correlation Coefficient (PCC) between wind speed and wind direction for the single races. PCC for two generic vectors $\mathbf{x} = (x_1, \ldots, x_n), \mathbf{y} = (y_1, \ldots, y_n)$ is defined in Equation 5.1

$$\text{PCC} = \frac{\sum_{i=1}^{n}(x_i - \bar{\mathbf{x}})(y_i - \bar{\mathbf{y}})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{\mathbf{x}})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{\mathbf{y}})^2}} \tag{5.1}$$

PCC can assume values between -1 and 1. When the absolute value of PCC is close to 1, it means that the vectors are strongly correlated (positively or negatively according to PCC's sign). Conversely, a PCC close to 0 indicates no correlation. The PCC values in Table 5.2 indicate poor or no correlation between wind speed and direction. Moreover, there is an almost even distribution of positive and negative PCC. For this reason, in this work two independent

Table 5.2: Correlation coefficients for wind speed and direction

Race	1	2	3	4	5	6	7	8	9
PCC	-0.088	-0.329	0.161	0.324	0.368	0.373	0.238	0.373	-0.014
Race	10	11	12	13	14	15	16	17	18
PCC	0.081	-0.701	-0.405	0.154	-0.475	0.234	-0.164	-0.0255	-0.018
Race	19	20	21	22	23				
PCC	-0.149	0.452	-0.145	-0.125	0.626				

models are used to forecast wind speed and direction.

5.2 Input and output

The recorded dataset is averaged over 10 seconds using 50 consecutive values, leading to a re-sampled dataset with one value every 10 seconds (0.1 Hz). This time step is selected based on the time of interest for tactical decisions in AC races. In fact, typically an AC72 takes 20 to 30 seconds (depending on the wind speed) to complete a tack and reach the maximum BS. If another tack was initiated during this period, this would significantly increase the time needed to reach again the maximum BS. Therefore wind shifts happening over a time period that is shorter than the time needed for a tack are unlikely to influence tacking decisions.
The method used for wind forecasting is based on the approximation expressed in Equations 5.2 and 5.3:

$$\hat{ws}(t+\Delta t) = \hat{f}_1(ws(t), ws(t-\Delta t), \ldots, ws(t-m\Delta t)) \tag{5.2}$$

$$\hat{wd}(t+\Delta t) = \hat{f}_2(wd(t), wd(t-\Delta t), \ldots, wd(t-m\Delta t)) \tag{5.3}$$

where $\Delta t = 30$s, $ws(t)$ and $wd(t)$ are the wind speed and wind direction, respectively, at time t and as in previous examples the hat refers to approximated quantities.

5.3 ANN model optimisation

5.3.1 Optimisation parameters

The chosen model for ANN wind forecasting is a feed-forward multi-layer perceptron (MLP). The parameters investigated are varied as follows:

1. Input length: the length of the input vector varies from 2 to 15 minutes in one-minute steps
2. Moving average: the input data are processed with a moving average of varying length, from 4 to 10 minutes in two-minutes step. The case of no data pre-processing is also considered.
3. ANN size: two layers of increasing size are used. Specifically, 15, 20, 25, 60 and 100 neurons per layer are used.

In Chapter 8 it will be shown that the optimal structure for wind speed forecasting consists in a two-layer perceptron that has 20 neurons per layer, taking as input 18 values, processed with a moving average over 4 minutes. The optimal model for wind direction forecasting consists in a two-layer perceptron that has 14 neurons per layer, taking as input 12 values, processed with a moving average over 4 minutes.

5.3.2 ANN-ensemble forecast

Because the synaptic weights are randomly set and then iteratively adjusted with the training of the network, networks initialised with different sets of random weights lead to slightly different output. Therefore, following the procedure outlined in Tagliaferri *et al.* (2015), what it is called *ANN forecast* in the following is the average of the outputs of one ensemble made of ten identical networks trained independently using different sets of initial synaptic weights. This averaging process decreases the chances of large errors due to unfortunate initialization-training combinations (Chaouachi and Nagasaka, 2012). The size of the ensemble is chosen arbitrarily and in Tagliaferri *et al.* (2015) it is shown that a larger ensemble can lead to a more accurate forecast but to higher computational time.

5.3.3 Evaluation criteria

Of the 23 days available, 22 are used for training and one is used for testing. To avoid a bias in the results due to the peculiarities of one specific day, the training and testing procedures are repeated 23 times, each time using a different day for testing the ANN. As a multi-step-ahead forecast is computed, the indices are time-dependent with respect to how far in the future the forecast values are. For instance, when the forecast values are $ws(t+\Delta t), ws(t+2\Delta t), \dots ws(t+n\Delta t)$, the indices will have the generic form $I_s(\Delta t), I_s(2\Delta t), \dots, I_s(n\Delta t)$. Similarly, the corresponding indices for wind direction forecasting will be noted as $I_d(\Delta t), I_d(2\Delta t), \dots, I_d(n\Delta t)$. The indices are always averaged over all the elements of all the testing sets. In the following, for simplicity the indices are defined using the wind speed case as an example, and the corresponding definitions for the wind direction case are omitted.

The forecasting performance of the ANN is evaluated using the following indices:

- Mean Absolute Error: $MAE_s(k\Delta t)$ is the average error on a k-steps-ahead wind speed forecast. It is obtained by averaging the absolute value differences between the recorded values and the corresponding values forecast by the ANN k time-steps before. For simplicity of notation, let $\mathscr{T}$ denote the testing set, i.e. the collection of values $ws(t+\Delta t), ws(t+2\Delta t), \dots ws(t+n\Delta t)$, and let $\hat{ws}(t+\Delta t), \hat{ws}(t+2\Delta t), \dots \hat{ws}(t+n\Delta t)$ be the corresponding forecast values. MAE is defined as follows:

$$MAE_s(k\Delta t) = \frac{1}{|\mathscr{T}|} \sum_{\mathscr{T}} |ws(t+k\Delta t) - \hat{ws}(t+k\Delta t)| \tag{5.4}$$

  The interpretation of MAE is straightforward: it represents, on average, the expected difference between real and forecast values.

- Mean Square Error: $MSE_s(k\Delta t)$ differs from the MAE because squaring the differences between real and forecast value leads to giving more importance to high differences and less importance to small differences. Using the same notation of Equation 5.4 it is defined as:
$$MSE_s(k\Delta t) = \frac{1}{|\mathscr{T}|}\sum_{\mathscr{T}}(ws(t+k\Delta t) - \hat{ws}(t+k\Delta t))^2 \tag{5.5}$$

- R^2: called also coefficient of determination in the literature, it represents the correlation between forecast and real values. The closer the value to 1, the better the forecast. It is defined as:
$$R_s^2 = \frac{cov_{\mathscr{T}}(\hat{ws}(t+\Delta t), ws(t+\Delta t))}{var(\hat{ws}(t+\Delta t))var(ws(t+\Delta t)} \tag{5.6}$$

- Mean Effectiveness Index: this index is defined in order to evaluate the forecasting ability of the ANN for the specific sailing application. A forecast can be considered "correct" if it leads to the right tactical choice. The MEI is defined to approximate this evaluation without needing to simulate a whole race (to check *a posteriori* if the decision taken was the right one).

  The Effectiveness Index (EI) is defined to evaluate the performance of the forecast for an arbitrary time window of two minutes ahead. In particular, we want to know if the average wind direction for one minute ahead will change by more than 3° and in which direction; and also if and in which direction it will change in the second minute with respect to the first one in order to understand if it is a temporary or a permanent wind shift. The threshold of 3° is chosen for an AC72-class yacht sailing an AC course. Typically, an AC72-class yacht takes about 20 seconds to complete a tack, and during the manoeuvre the average boat speed is approximately 70% of the optimum speed. Therefore, a tack leads to a loss of about 6s. Let us consider again the example shown in Figure 1.2. The black boat starts from the centre of the course, she sails for say one minute, then tacks and comes back to the centre of the course. Say she took two minutes plus 6 seconds for the tack. If the optimum course wind angle is 45°, in two minutes a shift of 3° would lead to an advantage of about 6 seconds on the grey boat. On the contrary, if the grey boat had tacked at the start and followed the black boat from the beginning, she would be 6 seconds behind the black boat due to the time spent to tack at the start. Therefore a wind shift of 3° can be considered a good approximation of the threshold at which the decision of tacking or sailing in the same direction has to be made.
  The effective index (EI) for a one-minute-ahead forecast is defined as follows:
  The mean effective index (MEI) is defined as the average of the EI:
$$MEI = \overline{EI} \tag{5.7}$$
  where the bar represents the average over the test set.

The MEI for the wind speed is defined in the same way as the MEI for the wind direction, and the threshold corresponding to the 3° threshold is at 0.5 m/s.

5.4 Comparison with other models

The performance of the optimal ANN structures is compared with three other forecasting models in order to quantify the advantages of using ANN.

5.4.1 ARMA

Auto-Regressive Moving Average (ARMA) is a standard model commonly used as a benchmark for forecasting techniques and for time series models. It has also been widely applied in wind forecasting (see Section 2.2). When using ARMA, the current value in a time series is computed as a linear combination of previous values (the auto-regressive term) plus a linear combination of independent random variables. Recalling Equation 2.5, the defining equation of an ARMA(p,q) process is:

$$\mathbf{w}(t) = \sum_{i=1}^{p} a_i \mathbf{w}(t-i) + \sum_{i=1}^{q} b_i \xi(t-i) + c \tag{5.8}$$

The standard technique for finding the coefficients to best fit the data is based on a maximum likelihood estimator, introduced in Section 2.2.2. This technique is implemented in many analytical software packages, and the one used in this work is the function *armax*.
The optimal p and q parameters are found by using the Bayesian Information Criterion (BIC). BIC was introduced by Schwarz (1978), and together with the Akaike (1973) information criterion it is the most used parameter to compare two models. When performing this comparison, the fitted model favored by the BIC is the candidate model which is *a posteriori* more probable, or the model which is most likely given the available dataset. BIC as well is based on the likelihood function.
Let's consider the problem of approximating a sequence of data y, when there are a set of candidate model functions $\hat{f}_1, \ldots, \hat{f}_L$. We assume that each $\hat{f}_1, \ldots, \hat{f}_L$, is characterised by a vector of parameters θ_k (recall the definitions in Section 2.2.2. In the ARMA case, θ_k is the vector containing the coefficients a_i, b_i, c.

$$BIC = -2 \ln \mathscr{L}(\theta_k | y) + k \ln n \tag{5.9}$$

The value of BIC is therefore a function of the likelihood for the parameters θ_k defining the model, but also of the dimension of the model.

5.4.2 Support Vector Machines

SVR model is implemented using the software LIBSVM (Chang and Lin, 2011), together with its Matlab interface. This software has already been used successfully in time series forecasting, and has the advantage of being fully documented making it suitable for the use as a simple benchmark (Chen *et al.*, 2004). The authors suggest the use of radial basis kernel function (RBF), because of the low number of the hyperparameters that influence the complexity of model selection. The radial basis function is shown in Equation (5.10):

$$k(\bar{x},\bar{y}) = \exp(\frac{||\bar{x}-\bar{y}||}{2\sigma^2}) \tag{5.10}$$

A "trial and error" approach is suggested for the identification of the best model parameter σ, which is followed in this work. An exponentially growing sequence of values for γ is used.
Similarly to the ANN case, a parameter study for different lengths of the input vector and lengths of moving average is presented. It will be shown that there is an optimum for each of these parameters that allows the maximum accuracy of the forecast.

5.5 Markov Chain model

Two kinds of information are needed by the routing algorithm: the first one is a punctual forecast, to compute expected positions for the yacht. The second one is a distribution, to compute the expected time needed to reach that position. The probability distribution is computed with the use of MC models, that also allow the use of coherent risk measures for risk modeling.

5.5.1 Medium-term wind modeling with Markov Chains

The Markov model described in this section follows the description appeared in Tagliaferri *et al.* (2014). As introduced in Section 4.4, a Markov model is defined through a transition matrix P, where the generic element of the matrix p_{ij} represents the probability of transition from state i to state j. The transition matrix P can be computed from a set of data by subdividing the data points in bins corresponding to different states, and using the following procedure. Let n_{ij} be the number of occurrences of subsequent values falling in bins $w_i w_j$ and let d_{ij} be the number of values falling in bin w_i. Then

$$p_{ij} = \frac{n_{ij}}{d_{ij}} \tag{5.11}$$

Figure 5.4 shows a graphical representation of the transition matrix for the Markov model. With a notation that will be used throughout this work, a grey scale is used to represent values in the interval $[0,1]$ where darker colours represent higher probabilities. As an example, the fifth row of the matrix is highlighted. The row represents the transition probabilities when the process is in state w_5. The figure shows how the darker colour corresponds to higher bar in the histogram representation of the transition probabilities. It can be noticed that the diagonal is dominant,

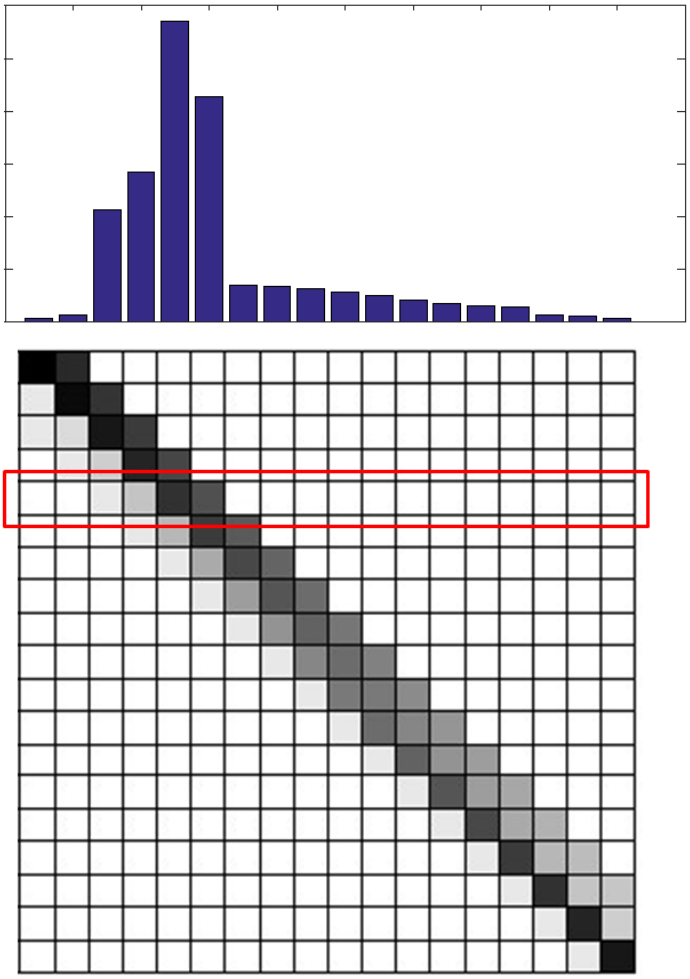

Figure 5.4: Representation of the transition matrix for the wind model.

meaning that, in general, if the wind is in state i, the most probable state for the next step is again state i. Moreover, when the wind has deviated from the mean, the event of a shift back towards the mean value is more likely than one in the same direction. In this case, time steps of 30s are used.

5.6 Artificial wind time series

MC can be used to generate artificial wind time series. These are needed in order to be able to test the algorithms on a high number of wind patterns. An artificial time series can be generated according to an MC model by using a random number generator. Let $z_0, z_1, z_2, \ldots$ be a series of independent identically distributed random variables, uniform on the interval $[0, 1]$. The initial

state X_0 is defined as $\omega_0 = w_i$ if

$$\begin{cases} \sum_{k=0}^{i} p_k^0 < z_0 \\ \sum_{k=0}^{i+1} p_k^0 > z_0 \end{cases} \tag{5.12}$$

Similarly, for the subsequent time steps, the value for the general ω_t, given the previous state $\omega_{t-1} = w_i$ is defined to be $\omega_t = \omega_j$ if

$$\begin{cases} \sum_{k=0}^{j} p_{ik} < z_i \\ \sum_{k=0}^{j+1} p_{ik} > z_i \end{cases} \tag{5.13}$$

Chapter 6

Velocity Prediction and aerodynamic interactions between two boats

6.1 Velocity Prediction Program

The VPP presented requires the geometry of the boat as input. The parameters needed are listed in Table 6.1, using the same notation as in Larsson *et al.* (2014). An advantage of the VPP presented is that the only quantities required are related to the geometry of the boat. Table 6.1 lists the input quantities required by the VPP. For simplicity, the quantities defining the appendages (rudder and keel) are defined only once. For instance, $R_{I,app}$, defines both the induced resistance of the keel ($R_{I,keel}$) and of the rudder ($R_{I,keel}$). The fundamental equations used for the VPP are listed in this Section.

6.1.1 Aerodynamic forces

The aerodynamic forces can be expressed in terms of the lift (L), which acts perpendicular to the mast and to the apparent wind direction and the drag (D), along the apparent wind direction. The aerodynamic model described in Larsson *et al.* (2014) is based on the work by Hazen (1980). The model used in the VPP presented here has a similar approach, and the regression formulas used are the ones used by the Offshore Racing Council in the VPP used for handicapping rules (ORC, 2013).
The values of L and D are directly proportional to the square of AWS, to the sail area, and and to lift and drag coefficient, respectively.

$$D = \frac{1}{2}\rho C_l AWS^2 A_N \tag{6.1}$$

$$L = \frac{1}{2}\rho C_d AWS^2 A_N \tag{6.2}$$

The lift and drag coefficients are defined for each sail a function of the apparent wind angle, and are presented in the Appendix (Tables A.5 and A.6). The global C_l and C_d are then computed as a weighted average on the areas.

$$C_l = \frac{C_{lM}A_M + C_{lF}A_F}{A_N} \tag{6.3}$$

Table 6.1: VPP input definition

Symbol	Description	Unit
A_N	Nominal sail area	m^2
A_w	Waterplan area	m^2
AWS	Apparent wind speed	m/s
AWA	Apparent wind angle	deg
BS	Boat speed	m/s
B_{wl}	Beam of the hull at the waterline	m
$C_{app,r}, C_{app,t}$	Root/Tip appendage chord	m
C_p	Prismatic coefficient	-
D_{app}	Draft of the appendage	m
D_{kg}	Total displacement	kg
D_c	Displacement of the canoe body	m^3
F_r	Froude number	-
g	Gravity acceleration	m/s^2
G_M	Metacentric height	m
G_Z	Arm of hydrostatic moment	m
h	Heel angle	deg
h_{rad}	Heel angle	rad
H_w	Significant wave height	m
L_{wl}	Length of the waterline	m
L_{CB}	Longitudinal centre of buoyancy, positive backwards	m
L_{CF}	Longitudinal centre of waterplan area, positive backwards	m
Re	Reynolds number	-
S_w	Wetted surface area of the hull	m^2
S_{wc}	Wetted surface of the canoe body	m^2
T	Draft of the hull	m
T_c	Draft of the canoe body from waterline	m
Th_{app}	Thickness of the appendage	m
TWA	True wind angle	deg
Z_{app}	Arm for the appendages forces	m
Z_C	Arm for the aerodynamic side force	m
Z_k	Distance from the centre of buoyancy of the keel to the bottom of the hull	m
Z_r	Distance from the centre of buoyancy tof the rudder to the bottom of the hull	m
(x_0, y_0, z_0)	Reference system	
(x_h, y_h, z_h)	Heeled reference system	
(F_x^A, F_y^A, M_h^A)	Aerodynamic forces and heeling moment	N, N, Nm
(F_x^H, F_y^H, M_r^A)	Hydrodynamic forces and righting moment	N, N, Nm

The nominal area (A_N) is defined as the sum of the areas of the single sails

$$A_N = A_M + A_F \tag{6.4}$$

When the area of the sails is not available among the input data, it can be computed as shown in Equations 6.5, 6.6, for mainsail and foresail respectively.

$$A_M = \frac{P}{8}\left(E + 2MGL + 2MGM + 1.5MGU + MGT + \frac{HB}{2}\right) \tag{6.5}$$

$$A_F = 0.1125 \cdot JL \cdot (1.445 \cdot LPG + 2 \cdot JGM + 1.5 \cdot JGU + JGT + 0.5 \cdot JH) \tag{6.6}$$

Figure 6.1 shows the definition of the quantities needed for the approximation of the mainsail and jib areas.

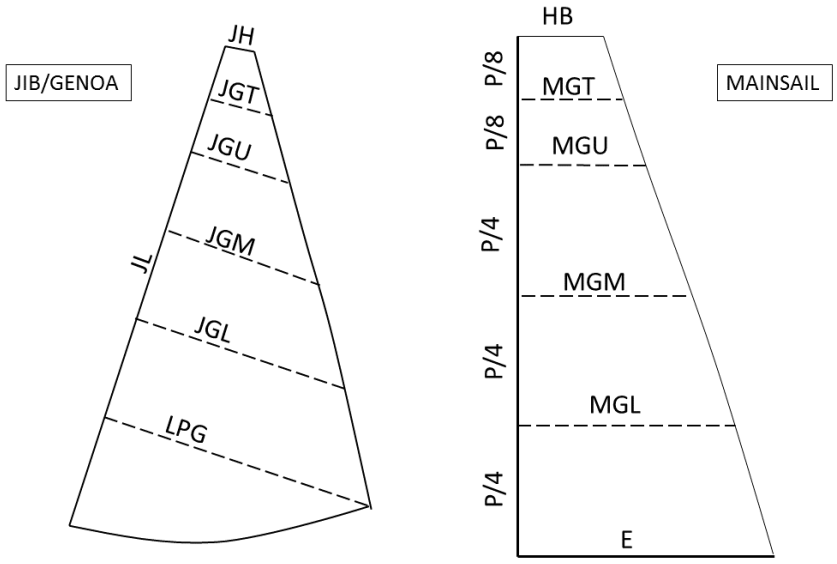

Figure 6.1: Definitions of sails parameters.

Following the approach in Kerwin and Newman (1979) and Hazen (1980), a flattening parameter ($F \in [0,1]$) is used as a scaling factor for the sail lift coefficient. If $F = 1$ the lift coefficient is the C_l computed as a function of the sail geometry. If $F < 1$ then the lift coefficient becomes $F \cdot C_l$. The drag coefficient is modified accordingly. In practical terms, having a flat lower than 1 corresponds to the common action of easing the sails sheets to sail at a lower heel angle when this leads to a higher boat speed. The VPP optimises F, which means that the VPP algorithm finds the F that yields the maximum BS.

6.1.2 Hydrodynamic forces

The hydrodynamic forces depend on the shape of the immersed parts of the yacht, on their roughness, and on the water flow. A first classification of these forces can be given according to their direction: the resistance, along the boat speed direction, and the side force, perpendicular to the boat speed.

Resistance

Equations 6.8-6.19 describe the computation of the different Resistance components as described in Keuning and Sonnenberg (1998). The dimensions of the hull used as input for the calculations are defined in Figure 6.2. The coefficients used in the Equations, which depend on the Froude number, are listed in the Appendix.

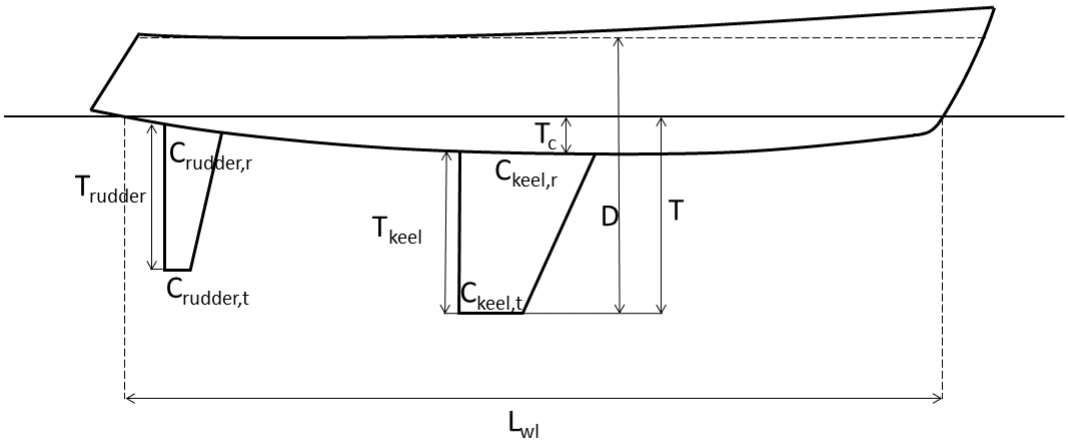

Figure 6.2: Definition of hull parameters.

- Viscous resistance: The viscous resistance is computed as follows

$$\frac{R_V}{\rho} = \frac{1}{2}\rho B S^2 S_w C_V, \;\; C_V = \frac{0.075}{(\log(Re)-2)^2} \tag{6.7}$$

- Residuary resistance. The residuary resistance of hull and appendages is computed using

Equations 6.8 and 6.9, respectively:

$$\frac{R_{R,hull}}{D_c\rho g} = c_1 + \left(c_2\frac{L_{CB}}{L_{wl}} + c_3 C_p + c_4\frac{D_c^{\frac{2}{3}}}{A_w} + c_5\frac{B_{wl}}{L_{wl}} \right)\frac{D_c^{\frac{1}{3}}}{L_{wl}} + \left(c_6\frac{D_c^{\frac{2}{3}}}{S_{wc}} + c_7\frac{L_{CB}}{L_{CF}} + c_8\left(\frac{L_{CB}}{L_{wl}}\right)^2 + c_9 C_p^2 \right)\frac{D_c^{\frac{1}{3}}}{L_{wl}} \tag{6.8}$$

$$\frac{R_{R,app}}{D_{app}\rho g} = c_1 + c_2\left(\frac{T_{app}}{B_{wl}}\right) + c_3\frac{T_c + Z_{app}}{D_{app}^{\frac{1}{3}}} + c_4\frac{D_c}{D_{app}} \tag{6.9}$$

- Heel resistance. Heel resistance denotes the variation of residuary resistance due to the fact that the boat sails at a heeled angle. The change in resistance for the hull is computed for a reference heel angle of 20° using Equation 6.10. This change is then transformed for a generic heel angle by using Equation 6.11. Equations 6.12 and 6.13 show the corresponding formulas for the appendages:

$$\frac{\Delta R}{D_c\rho g} = u_1 + u_2\frac{L_{wl}}{B_w l}T_c + u_3\frac{B_{wl}}{T_c} + u_4\left(\frac{B_{wl}}{T_c}\right)^2 + u_5\frac{L_{CB}}{L_{wl}} + u_6 L_{CB}^2, \tag{6.10}$$

$$R_{R,hull,h} = 6\Delta R h^{1.7} \tag{6.11}$$

$$R_{R,app,h} = D_{app}\rho g C F_r^2 h_{rad} \tag{6.12}$$

$$C = -3.5837\frac{T_c}{T} - 0.0518\frac{B_{wl}}{T_c} + 0.5958\frac{T_c}{T}\frac{B_{wl}}{T_c} + 0.2055\frac{L_{wl}}{D_{app}^{\frac{1}{3}}} \tag{6.13}$$

- Induced resistance.

$$R_{I,app} = \frac{S_F^2}{\frac{1}{2}\rho\pi T_{eff}^2 B S^2\cos^2(h)} \tag{6.14}$$

$$T_{eff} = \left[\left(c_1\frac{T_c}{T} + c_2\frac{T_c^2}{T} + c_3\frac{B_{wl}}{T_c} + c_4 T_{ratio}\right)(c_5 + c_6 F_r)\right] T \tag{6.15}$$

$$T_{ratio} = \frac{C_{app,t}}{C_{app,r}} \tag{6.16}$$

and for the wave resistance

$$R_W = \rho g L_{wl} H_w^2 a\left[100 D_c^{\frac{1}{3}}\left(\frac{k_{yy}}{L_{wl}}\right)\right]^b \tag{6.17}$$

$$k_{yy} = \sqrt{\left(\frac{I_{yy}}{m}\right)}\, I_{yy} = \sum_i m_i x_i^2 \tag{6.18}$$

where I_{yy} is the mass momentum of inertia and m_i are the masses of all components of the boat with the corresponding relative arms x_i. Following Larsson *et al.* (2014) k_{yy} can be considered as one quarter of the hull length.

The total resistance is then computed as the sum of the various components:

$$R = R_v + R_{R,hull} + R_{R,keel} + R_{R,rudder} + R_{R,hull,h} + R_{R,keel,h} + R_{R,rudder,h} + R_{I,keel} + R_{I,rudder} + R_W \tag{6.19}$$

Side force

As described by Garrett and Wilkie (1987), the side force is mainly due to the appendages

$$S_{F,app} = \frac{1}{2} C_l \rho B S^2 T_{app} C_m \tag{6.20}$$

where C_m is the chord of the corresponding appendage (keel or rudder).

Righting moment

To complete the hydrodynamic model of a sailing yacht, it is necessary to compute therighting moments. The hydrostatic moment RM depends upon the metacentric height GM and the arm of the hydrostatic moment GZ.

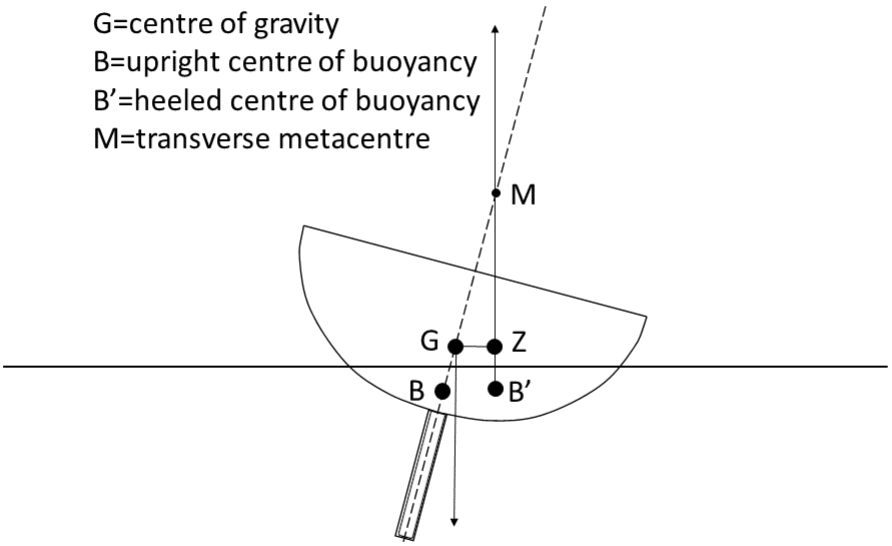

Figure 6.3: Righting moment.

6.1.3 VPP algorithm

In order to obtain a fast execution time of the VPP an optimisation routine which maximises BS as a function of the flattening parameter F, with fixed TWA and TWS was implemented. The optimisation problem is formulated in the following Equations

$$\max_{F\in[0,1]} BS(F) \tag{6.21}$$

$$F_x^A - F_x^H = 0 \tag{6.22}$$

$$F_y^A - F_y^H = 0 \tag{6.23}$$

$$M_h^A - M_r^H = 0 \tag{6.24}$$

The problem can be solved as a nonlinear constraint optimisation problem. The algorithm is implemented using an off-the-shelf Matlab function based on active-set algorithm. This optimisation algorithm can compute large steps to accelerate convergence, and it has good performances with non-smooth constraints. It divides the original problem in a sequence of unconstrained subproblems, solved at each iteration, whose solution converges to the solution of the original problem.

Another VPP based on an exhaustive search was implemented to verify the results of the optimisation-based VPP. Figure 6.4 shows a diagram describing the structure of this implementation of the VPP. The equilibrium in the three degrees of freedom is obtained via an exhaustive search, starting from some arbitrary initial guesses for the values of BS, λ and h. The forces are computed for these initial guesses, and then three concentric loops are used to converge to the equilibrium. The equilibrium is computed for a vector of possible values of the flat, and then the flat corresponding to the highest BS is chosen as the optimal one. This procedure is repeated for all the TWS and TWA required to obtain the polars. This implementation has the advantage of being effective, and of converging to the equilibrium with a user-defined level of accuracy. However the presence of three nested loops can lead to a very slow convergence.

Figure 6.5 shows the differences between the two VPP implementations in terms of computational time. The optimisation-based VPP takes on average 12.6% of the time needed by the loop-based VPP.

Figure 6.6 shows the output generated by the two VPP on a test case corresponding to the Y40yacht described in Larsson *et al.* (2014). The differences between the two results are of the order of the increment steps used in the loop-based VPP.

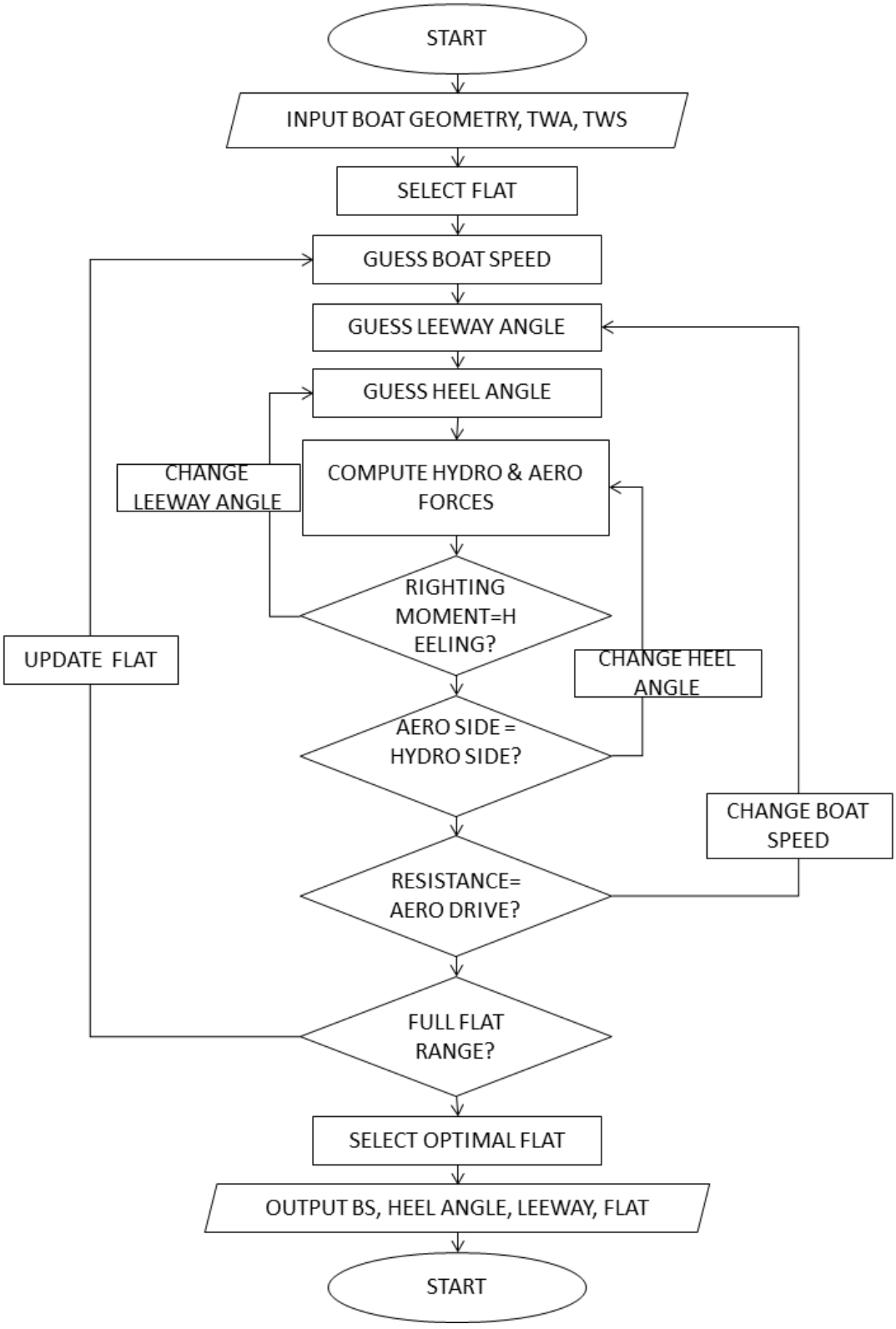

Figure 6.4: Structure of the loop-based VPP

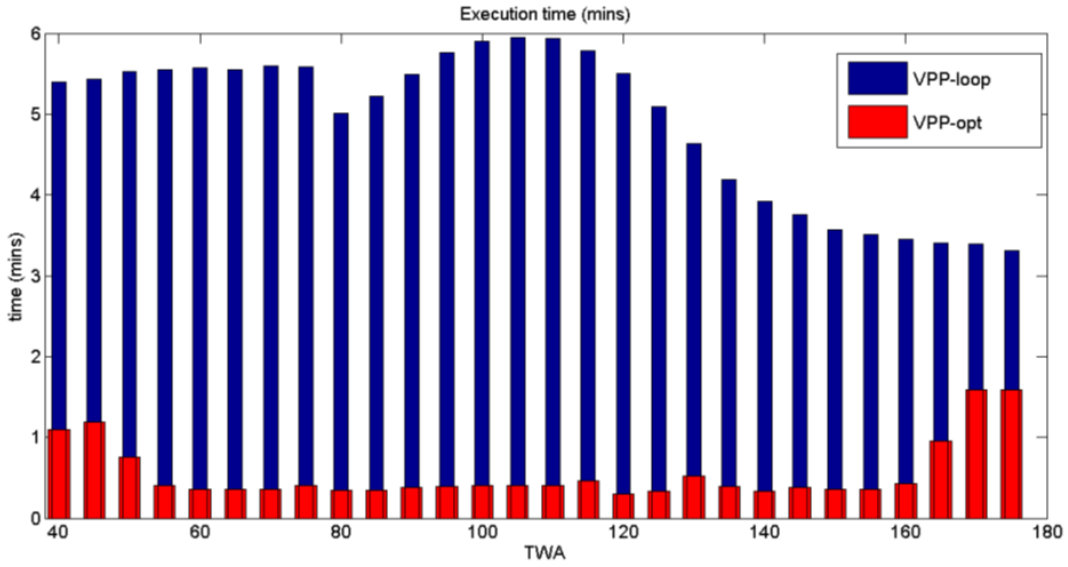

Figure 6.5: Comparison between VPP implementation: computation time

6.2 Aerodynamic interactions between two boats

The model for physical interactions between the two yachts competing in a match race is based on the experimental results of wind tunnel tests presented in Richards *et al.* (2013). These results are based on a series of wind tunnel experiments carried out in the Twisted Flow Wind Tunnel facility of Auckland University. The complete set of experiments is described in Aubin (2013).

Figure 6.7 (Aubin, 2013) shows an example of experimental results for AWA=5°, corresponding to circa TWA=25°. The yacht of interest is fixed, and the first set of measurement is taken with no interfering yacht. Then a second yacht is located at various radii on order to obtain a map of the interference as a function of the relative position of the yachts. The experimental data are then included in a VPP in order to compute variations in BS from the measured velocities. These speed variations can be presented as a polar diagram such as the one presented in Figure 6.8.

In order to include these results in the routing algorithm, the data from Aubin (2013) are interpolated as a function of TWS, TWA and position of the yachts. The choice of using the TWA as a parameter is due to the fact that the BS is not computed at every step of the computation. In fact, as the only information that is used by the routing algorithm (cf. Chapter 7)is the function $BS(TWA, TWS)$, an efficient way of including the influence of another yacht is to modify the polar diagram of the boat based on the position of the competitor.

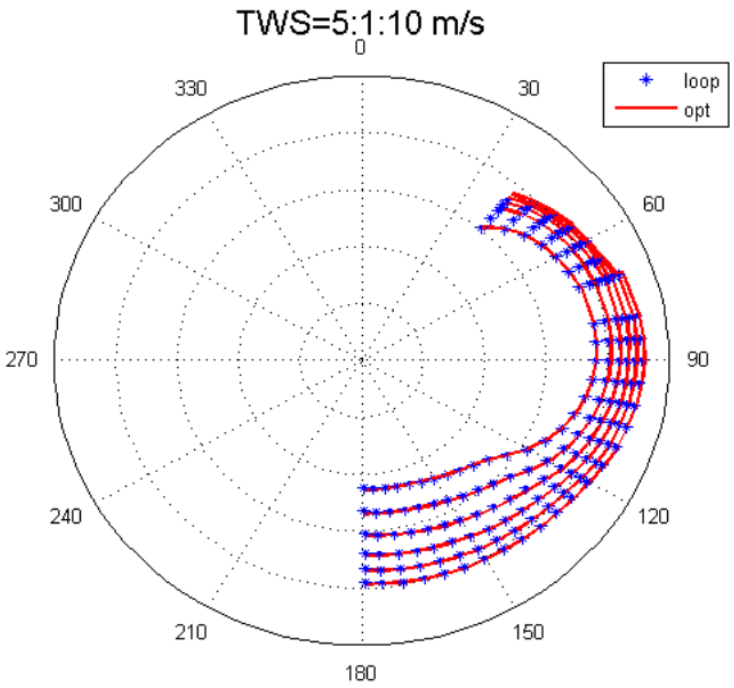

Figure 6.6: Comparison between VPP implementation: results

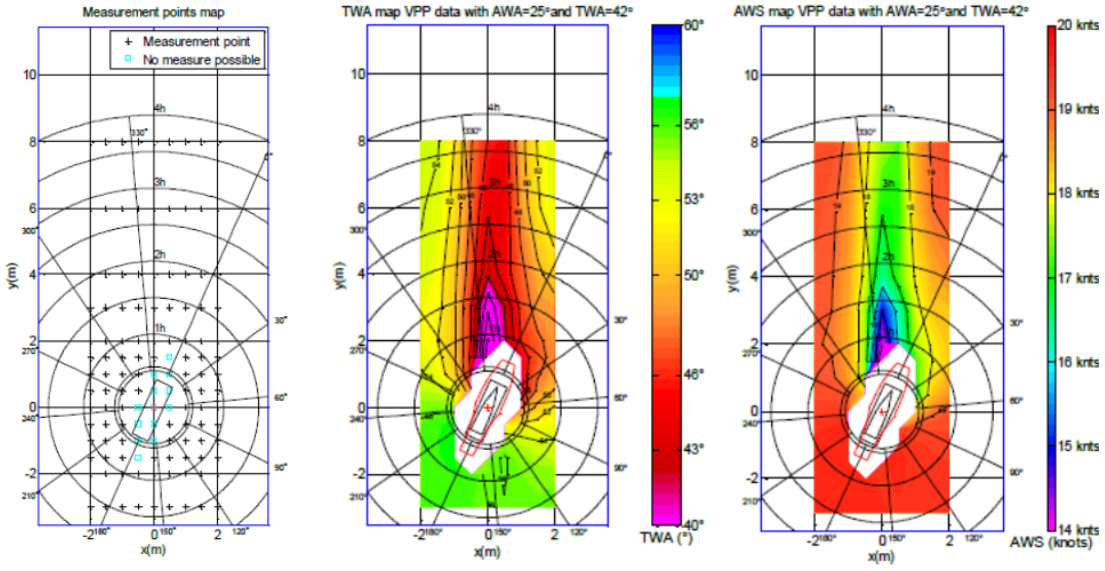

Figure 6.7: Wind tunnel test for multiple yachts interactions. Figure from Aubin (2013)

Let BS^* be the boat speed in presence of an influencing yacht. The first step of the interpolation is to compute the AWS and AWA as a function of TWS, TWA, BS. Let A be the yacht of interest, B the influencing yacht, and d_{AB} be the vector distance between the two yachts. The

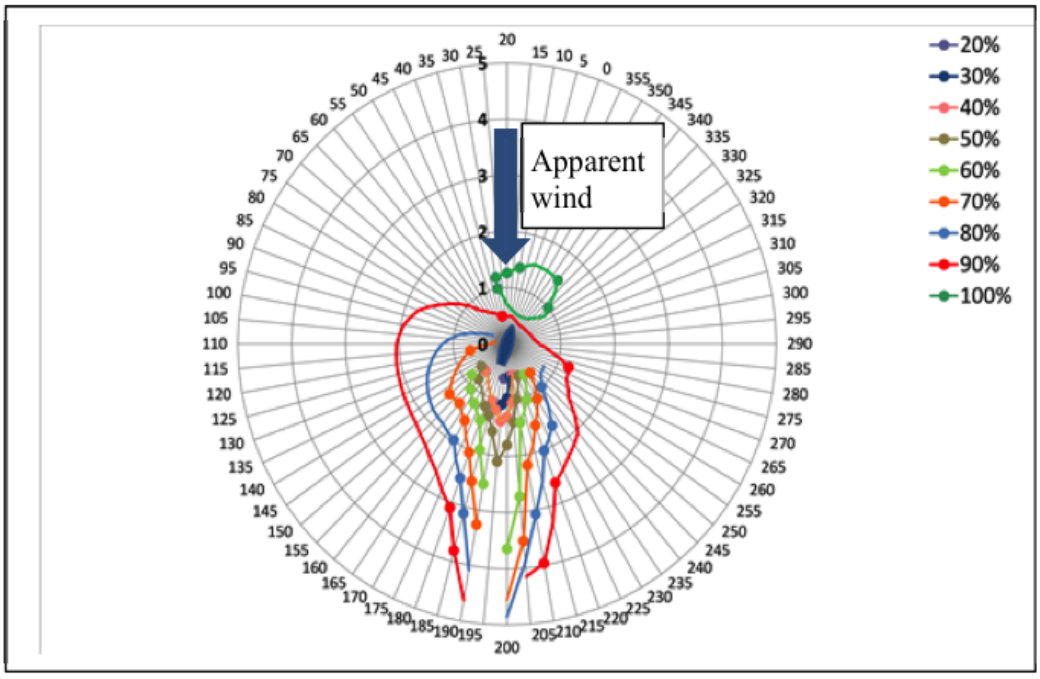

Figure 6.8: Boat speed variations with a fixed interfering yacht. Figure from Aubin (2013)

corrected BS is then computed as in Equation 6.25:

$$BS^*(TWS, TWA, BS, d_{AB}) = BS - f_{int}(AWA(TWS, TWA, BS), AWS(TWS, TWA, BS)) \quad (6.25)$$

where f_{int} represent the interpolating polynomial for the BS variations based on the coefficients presented in Aubin (2013) . The new BS^* substitutes BS in the computation of the grid in the proximity of the boat. The opponent is always assumed to follow the same course for n time steps. In this thesis $n = 5$.

Chapter 7

Routing Algorithms

7.1 Grid generation

Figure 7.1 shows the boundaries of the race area used in this work. The dimensions used are inspired by the typical length of a 35^{th} AC race area. The distance between the starting point and the upwind mark is of 5000 m, and the width of the area is 3000m. The course is assumed to be aligned with an average initial wind direction, which constitutes the *reference wind direction* RWD, and is a constant for the entire race. The area as shown in the Figure is delimited by ideal laylines, but in some cases, as shown in Chapter 9, the actual routes goes beyond those lines. There is a limited tolerance (100m) on the side boundaries for ease of grid computation.

The DP algorithm is based on a shortest path problem defined on a set of nodes connected

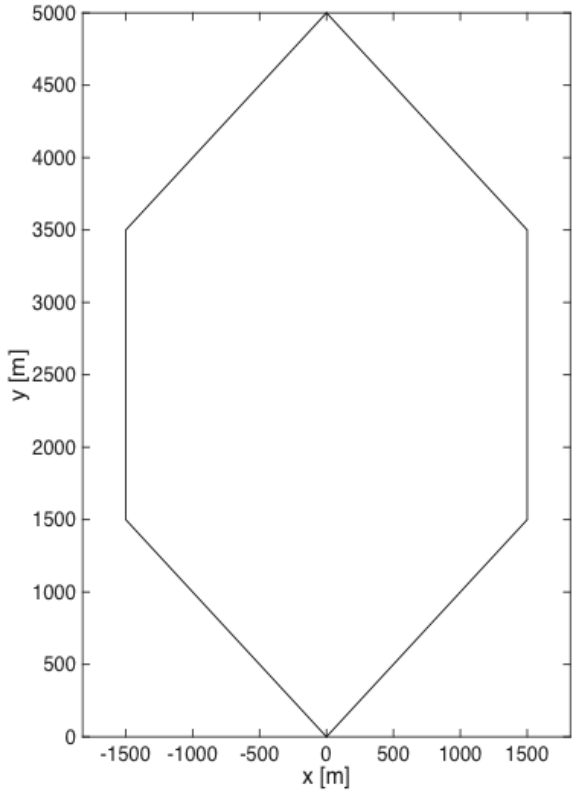

Figure 7.1: Racing area

in a graph, similar to the simplified example in Section 4.2. The set of nodes is not fixed, but their position depends on the wind forecast. Before formally describing the process of grid definition, an example to motivate this choice will be shown.

Let us consider the final phase of the upwind leg when the boat is reaching the mark in the case of a gradual wind shift towards the left. Figure 7.2 shows the optimal route towards the mark with two different underlying grids. In the left grid, the optimal route does not go through the nodes defined by the grid, therefore a certain approximation in the DP algorithm is needed. Conversely, the grid on the right shows an exact superposition of the route and one of the lines

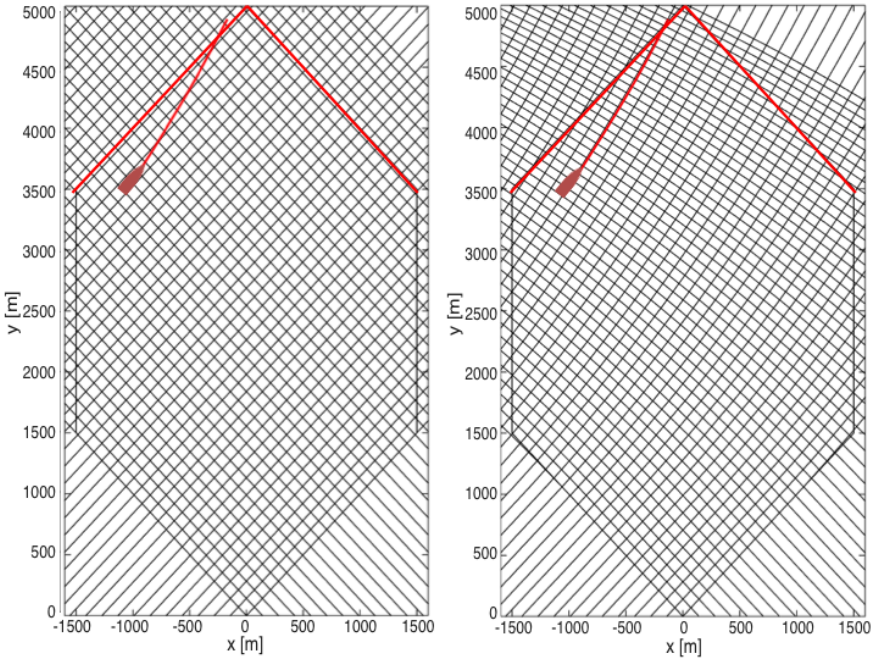

Figure 7.2: Comparison between grid with fixed spatial steps (left) and with wind-dependent steps (right).

constituting the grid. Ideally, the nodes defined by the grid should correspond to the reachable points on the racing area. Of course the racing area is a continuum, so every point within the race boundaries is always reachable, but the discretisation should be developed so that, if the yacht is in a given node belonging to the set of nodes defined by the grid, then the neighbour nodes should be reachable from that node. This property is not satisfied by the left grid in Fig. 7.2, but it is satisfied by the right grid, as shown by the red path followed by the yacht to reach the upwind mark. A curvilinear grid that matches the optimal route can be drawn if the future wind evolution is known. The grid defining the graph used for the DP algorithm is therefore based on the wind forecast, in order to predict the possible reachable points. The grid is then recomputed every time step. The main assumption underlying the construction of the grid is that, in the absence of tactical interactions due to the presence of a competitor, a skipper will always sail at maximum VMG. Figure 7.3 shows how to build the subsequent grid points given an initial node of coordinates (x,y). If the forecast wind when reaching the point (x,y) is represented by the wind w, then the possible reachable points in a given time step dt have

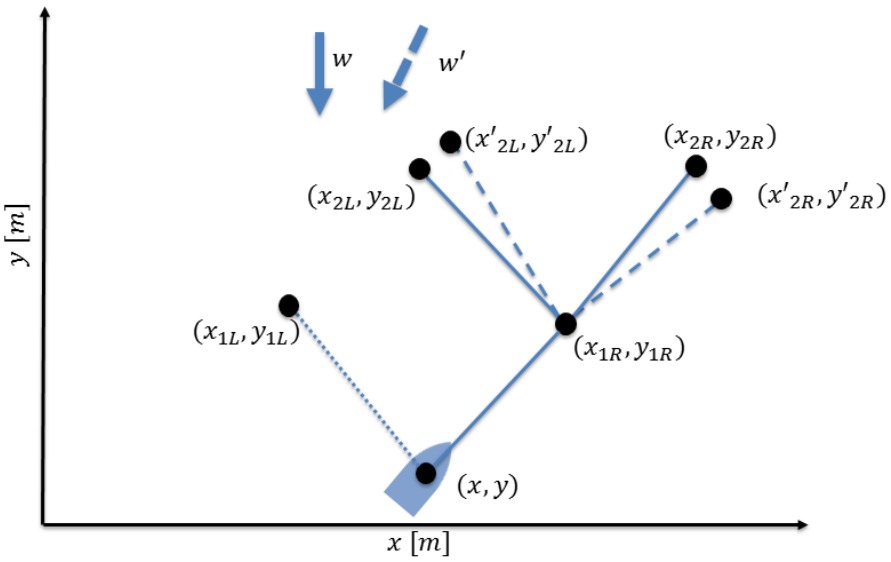

Figure 7.3: Construction of grid points.

coordinates $(x_{1L}, y_{1L}), (x_{1R}, y_{1R})$ depending on the current tack. Let us assume that the boat is on a port tack. Then in a period of time of $2dt$ the points $(x_{2L}, y_{2L}), (x_{2R}, y_{2R})$ can be reached. w is the wind which is *expected* at the moment when the grid is generated. A subsequent forecast could predict a different wind (e.g. w' in Figure 7.3), in which case the reachable points become $(x'_{2L}, y'_{2L}), (x'_{2R}, y'_{2R})$. This is why the grid construction is updated at every step.

If every node generated two subsequent nodes, the size of the grid would grow exponentially at each iteration. Rather than building the grid point by point, the grid is therefore built by defining a set of lines and then considering their intersections as the nodes constituting the graph underlying the DP algorithm. Figure 7.4 shows the construction of the initial grid. A set of M_0 evenly spaced point is defined on the x axis, where M_0 depend on the desired grid resolution. At step one of the computation the following operations are performed:

1. Yacht position: $(x_0, y_0) = (0, 0)$
2. Generate wind speed and direction forecast
3. Compute grid lines
4. Compute lines intersections. These points constitute the DP nodes
5. Store grid points in matrices G_x, G_y

The distance between grid points depends on the chosen time step. The step used in most of the simulations for this work is $dt = 5$s. This is the time step used in the DP algorithm and does not necessarily correspond to the time step used for the wind forecast.

The grid is built starting from the current position of the boat. The objective of the boat is to round the upwind mark clockwise. The mark itself is not necessarily a point of the grid. However, by construction the mark will lay between four grid nodes defining a grid cell. The leftmost node is considered the arrival node at each iteration.

At each step k of the computation the grid is re-computed. The current position of the boat,

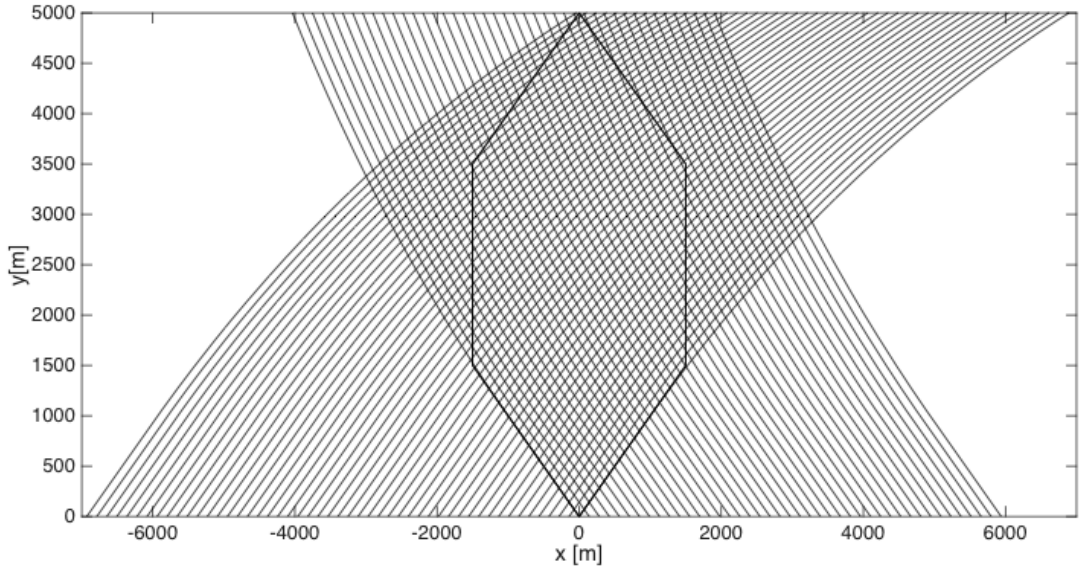

Figure 7.4: Example of grid construction.

(x_k, y_k) becomes the initial node. The equivalent of the initial points laying on the x axis are now a set of points evenly spaced (according to wind speed as for the first step) laying on the line of equation

$$y = \tan(CWA)(x - x_k) + y_k \tag{7.1}$$

If the CWA is zero, then the line generated is an horizontal line as shown in Figure 7.5, which is then followed by the grid construction for the new wind forecast.

7.2 Tide

The tidal current is considered to be a known function of time and space. In fact, sailors have commonly access to tidal current databases with spatial resolution usually of the order of some meters and time resolution of the order of minutes.
For this work, tidal data based on Tidetech commercial data have been used (Tid). Figure 7.6 shows an example of tidal current in San Francisco Bay. The dataset consists of sequences of grib files containing the latitudinal and longitudinal components of current velocity, and Figure 7.7 shows how the wind perceived by a moving yacht is affected by the presence of a current.

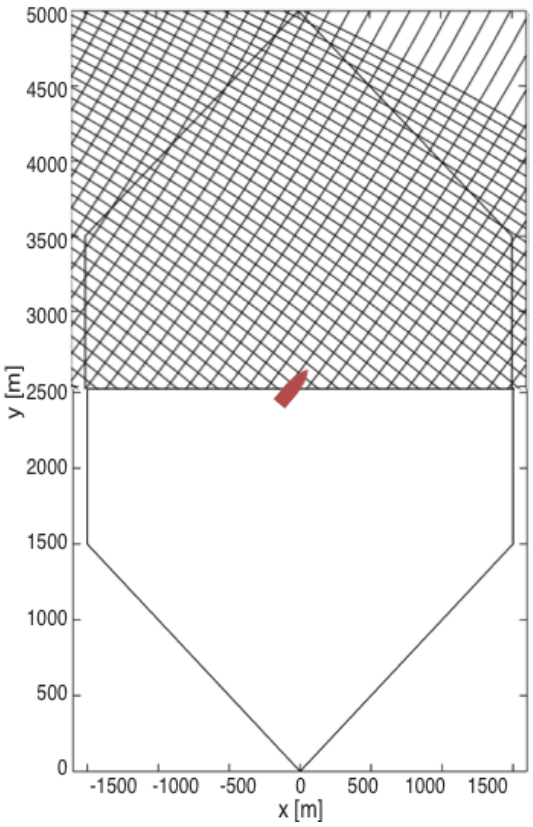

Figure 7.5: Example of grid construction.

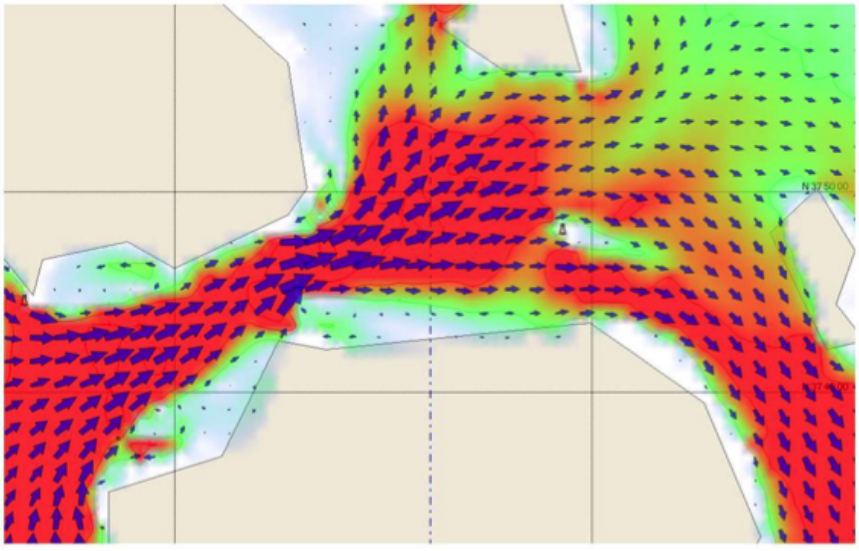

Figure 7.6: Example of tidal current data available to sailors

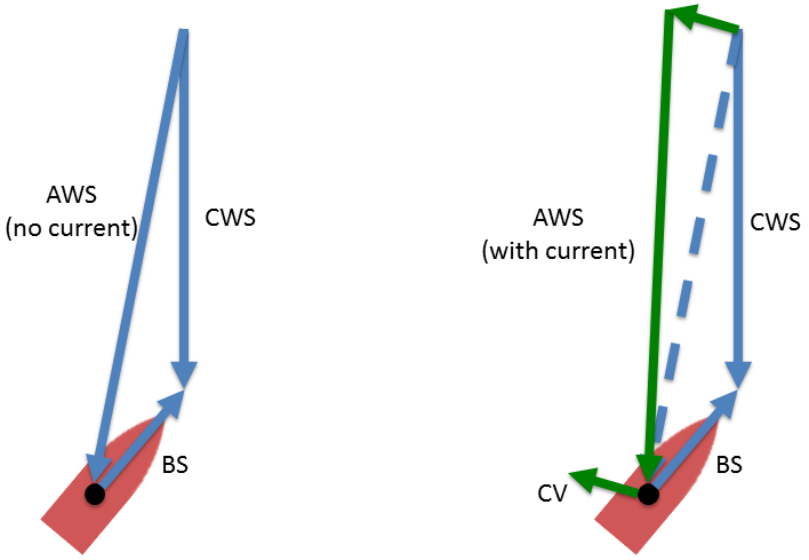

Figure 7.7: Wind triangle taking current into account.

7.3 Tacking

The key problem when computing a winning strategy is the time loss associated to tacking and jibing manoeuvres. Different time penalties can lead to dramatic changes in the computed optimal decision. Trivially, if there is no tack penalty a winning strategy allows a tack for every wind shift. Conversely, in any wind condition if the tacking penalty is higher than a certain threshold then the optimal strategy is the one allowing just one tack halfway through the upwind leg (and the same applies to jibing during a downwind leg).

Figure 7.8 shows two examples of tacks performed during the same race by ETNZ during the LV Cup. The speed during the manoeuvre is shown for intervals of 5s and the plot is aligned with the average wind direction. The tacking penalty used in this work is defined as a function of wind speed, giving a higher penalty for low wind speed. This leads to the general consequence of choosing, when this choice is possible, to perform a tack when the forecast wind speed is higher. For a constant wind speed of 10 m/s, in Tagliaferri *et al.* (2014) a penalty corresponding to sailing for 20s at 80% of VMG was used. The penalty for a wind speed of 5 m/s is set as sailing at 30s at 75% of the VMG. The penalties for other wind speeds are linearly extrapolated.

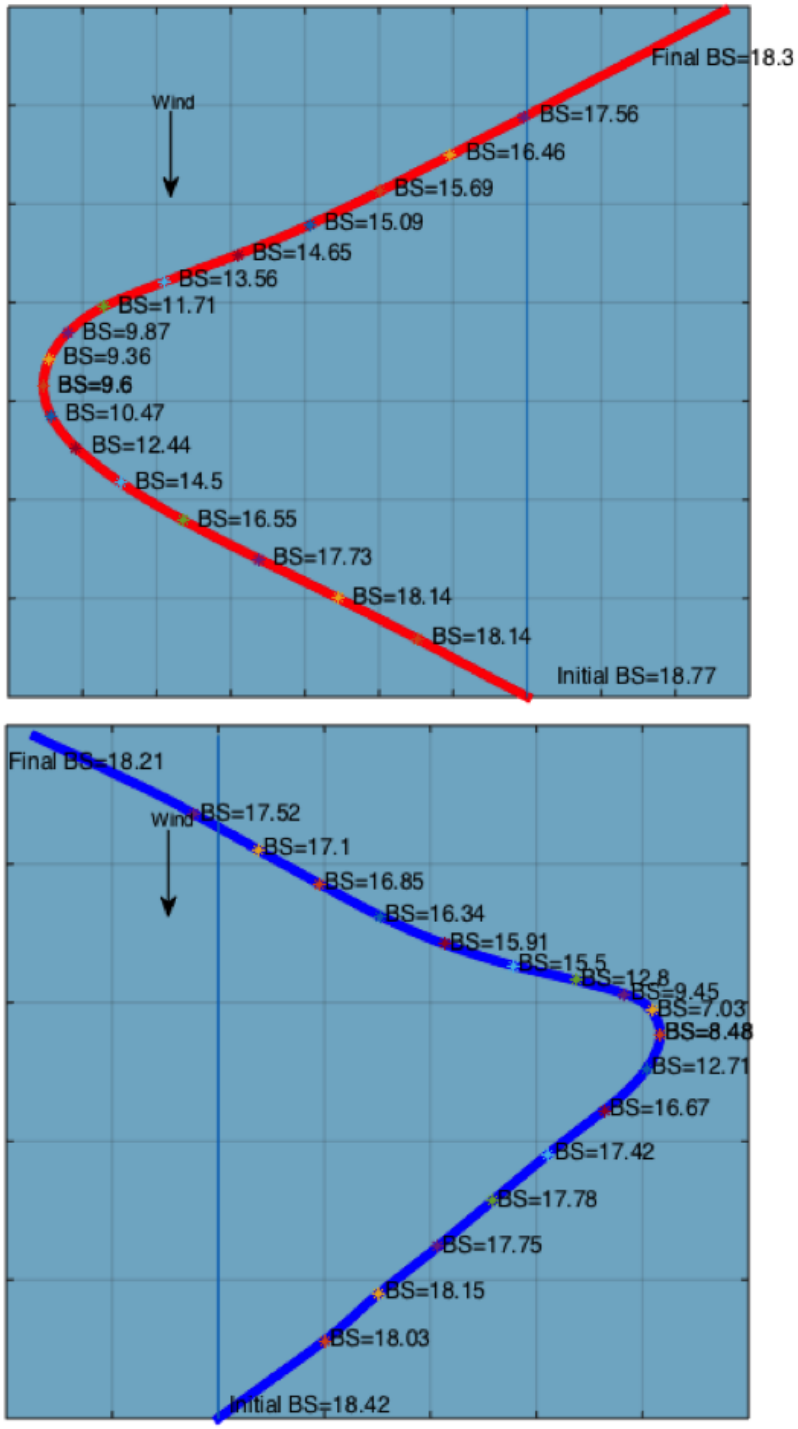

Figure 7.8: Examples of tack trajectories.

7.4 Risk modeling

The first risk investigation is carried out on the basic model. Coherent risk measures are used to simulate a risk-seeking and a risk-averse attitude. This procedure has been published in Tagliaferri *et al.* (2014), and is applied on the MC-based wind model.

The modelled risk attitudes have a double motivation. Consider first a skipper who is winning. She will try to behave safely, trying to stay ahead and minimise her losses in bad wind outcomes. In this context, she will be pessimistic about the upcoming wind shifts, assigning higher probability to the worst outcomes (i.e. heading shifts). Being pessimistic about random outcomes reduces risk, at some loss in expected performance.

Risk seeking behaviour has been less well studied, although it is often given as an explanation for participation in lotteries and negative expectation gambles, where optimistic participants place greater weight on winning probabilities than their real values. In our context we model risk seeking by choosing the best possible transition probabilities from a convex set of transition matrices. This has the following interpretation. A boat skipper who is losing will seek risk. If she adopts a minimum expected finish time strategy against another skipper who minimises his expected time to finish, then she will tend to make the same decisions (unless the boats see very different winds) and lose the race almost certainly. She will instead seek different wind conditions from the competitor, being optimistic about the possible advantageous wind shifts and assigning a higher probability to these outcomes (i.e. lifting shifts). Being optimistic about random outcomes increases risk, as well as incurring some loss in expected performance.

These concepts are implemented in the recursion by adding a transformation in the solver that post multiplies the transition matrix by another matrix which redistributes the probabilities. The resulting matrix has to be normalised in order to represent again a probability distribution. A sailor who is losing will seek risk. This corresponds to increasing her confidence of a lifting wind shift while discounting the likelihood of a heading wind shift. The transition matrices used to represent an optimistic skipper are shown in Figure 7.9. Advantageous shifts (cells below the diagonal when the skipper is to the left of the opposition, and cells above when on the right) happen with higher probability than in the risk-neutral case. The remaining probabilities in each row are reduced to add to one.

The transition matrices for a pessimistic, risk-averse skipper are constructed similarly. Here bad wind shifts (above the diagonal when the skipper is to the left of the opposition, and below the diagonal when on the right) happen with higher probability than in the risk-neutral case. The transition matrices for a risk-averse policy can be obtained by simply swapping the matrices in Figure 7.9.

The risk-neutral, optimistic and pessimistic matrices are selected according to the relative position of the skipper with respect to the opponent. Let T_d denote the expected time difference between two racing boats A and B.

$$T_d = \mathbb{E}(T(B) - T(A)) \tag{7.2}$$

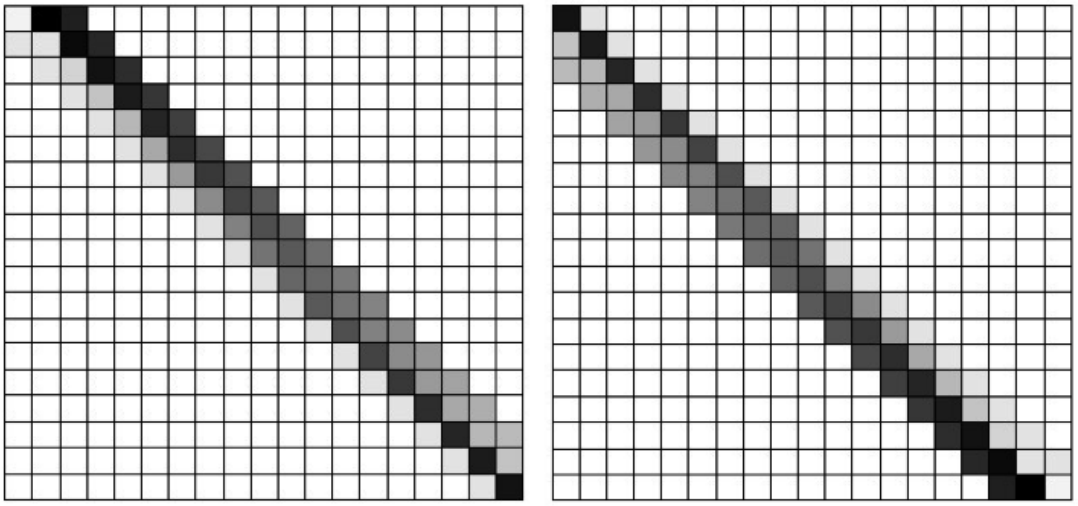

Figure 7.9: Modified transition matrices for a risk-seeking skipper. Advantageous wind shifts occur with higher probability than disadvantageous ones. (Left) Yacht on the left-hand side of competitor and (Right) yacht on the right-hand side of competitor.

where $T(A)$ and $T(B)$ represent the time needed by A and B, respectively, to complete the race. This function can be approximated by computing the expected time needed by the boat that is further behind to reach the cross-section of the race area where the leading boat is. Let T_{switch} be an arbitrary time difference. then if $T_d > T_{switch}$ then boat A is leading by a certain margin, and she uses the conservative strategy. Conversely, if $T_d < -T_{switch}$, then boat A is losing by a certain margin, and she uses the risk-seeking strategy.
A first set of experiment is carried out using the two modified matrices in Figure 7.9 and with $T_{switch} = 15s$. The optimal value for T_{switch} is then investigated together with the optimal modified matrices.

7.5 Dynamic routing

The proposed algorithm takes into account all the factors presented so far. The preliminary computations are based on the wind recorded before the race at the race location. This is used to compute the MC transition matrix and to train the two ANN models, for wind speed and wind direction.
When the race starts, the ANN forecast is used to generate the grid underlying the DP algorithm. The boat speed used for the grid computation is the corrected BS^* for the first five time steps, and then the rest of the grid is built by using the normal BS.

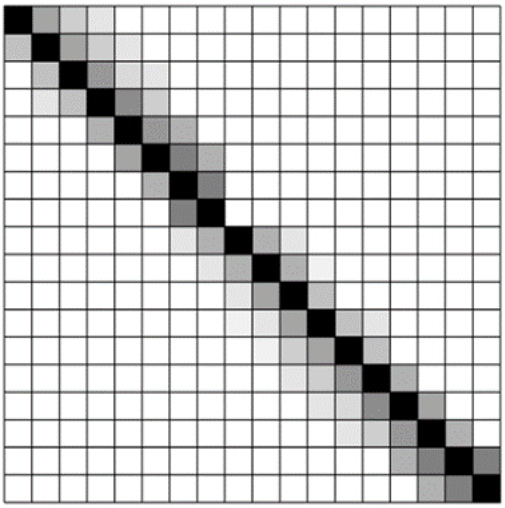

Figure 7.10: Transformation matrix for risk-seeking attitude

The stochastic DP algorithm is then executed minimising the problem

$$\mathbb{E}(C(U^{opt})) = \min_{U \in \mathscr{U}} \mathbb{E}(C(U, \omega)) \tag{7.3}$$

by finding a policy $U^{opt} = u_0^{opt}, \cdots, u_{N-1}^{opt}$. The element u_k^{opt} is the optimal decision to be taken at step k, i.e. the node to be reached in the previously built grid.
A set of rules aimed at avoiding collisions between the boats and at respecting the racing rules are implemented. In particular:

1. If the two boats meet, then the boat on a port tack increases the TWA, passing behind the other boat.
2. A boat cannot tack if this leads to its track crossing the opponent's under a certain fixed safety distance.

The safety distance is defined noting that the boats are modelled as points. The longitudinal safety distance is 10 m, the side distance is 5m.
The computations for the manoeuvre of bearing away and passing behind the opponent's boat is carried out by adding a node to the set of reachable nodes. This temporarily modifies the assumption that a boat always sails at maximum VMG. In the example shown in Figure 7.11, the red yacht expects to meet the opponent at the node indicated by the red dot. The black node is therefore added to the set of the reachable points. The opponent is assumed to follow a strategy aimed at minimising the expected time to complete the race. This means that he is expected to follow a reasonable route that depends on the forecast wind. To compute an optimal strategy, it may be possible to forecast the future position of the opponent, to take into account that the two boats might meet further in the future. However, in order to properly take into account such events, the computation of a probability distribution is required. In fact, let's

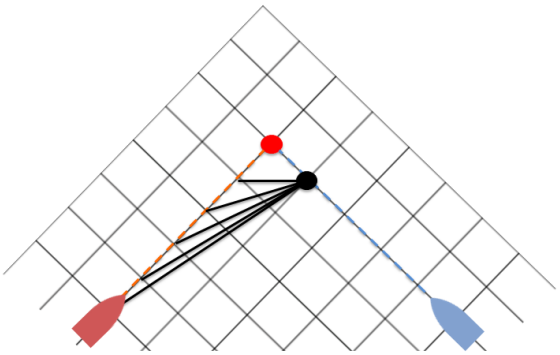

Figure 7.11: When boats meet on opposite tacks.

assume that with the wind conditions forecast at the beginning of the race the two boats can compute an optimal strategy which will lead them to meet in proximity of the upwind mark. This event will actually happen with a probability which is equal to the probability that the wind realisation is exactly the one forecast at the beginning and that the boats actually follow the computed strategy. The further this event is in the future, the closer this probability is to zero. In the current software implementation, the future window is set at one minute, which is the time frame at which the wind forecast has an average error lower than 2^{circ} (as shown in Chapter 8) and because a yacht is expected to perform not more than one tack in one minute.
An important hypothesis, not necessarily corresponding to reality, is that no yacht will take a decision that leads to a higher expected time with the aim of slowing the other yacht down.
To conclude the Method part, Figure 7.12 shows a flow chart of the steps used for the final algorithm.

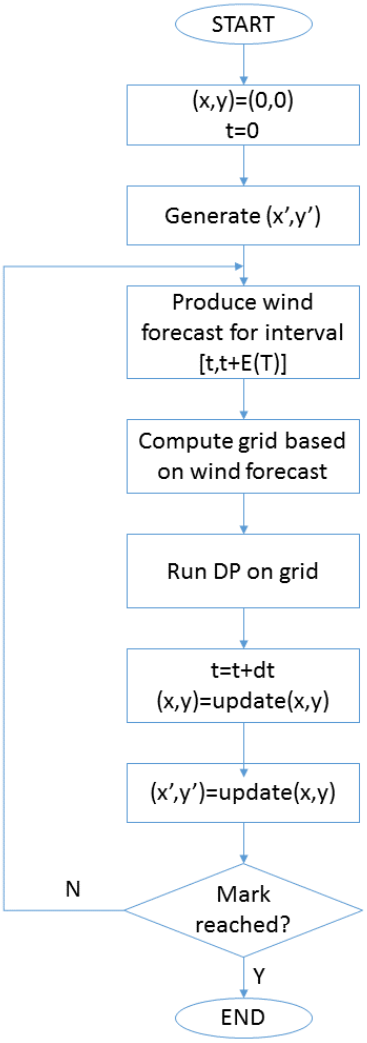

Figure 7.12: Summary of methodology used.

PART III

Results

Chapter 8

Wind forecasting

In this Chapter the results on wind forecasting with Artificial Neural Networks will be presented. The first section is dedicated to the optimisation of the ANN model, including ANN structure and input/output choice, for both wind speed and wind direction forecasting. The second section is focussed on the analysis of the performance of the optimised model. The chapter is concluded by a section on verification and validation of the model.

8.1 ANN structure optimisation

8.1.1 Wind direction forecasting

The optimisation is carried out on the test matrix defined in Section 5.3. The results presented in this Section are published in Tagliaferri *et al.* (2015). A two-step ahead forecast is performed using a feed forward ANN with two hidden layers containing an equal number of neurons.
Figures 8.1-8.6 present MAE and MEI of one and two minutes ahead for different lengths of the moving average, different lengths of the input vector, and different number of neurons in the hidden layers. Due to the multidimensionality of the test matrix, the trend of the two performance indices versus each parameter is shown only for the best combination of the other two parameters.
Figures 8.1 and 8.2 show the MAE and the MEI as functions of the length of the moving averages, where each hidden layer has 20 neurons and the size of the input vector is 18 (9 minutes). Figure 8.1 shows that for a moving average over 6 minutes, the MAE of one minute ahead is $1.7° \pm 0.3°$, while the MAE of the two-minute-ahead forecast is $3.0° \pm 0.2°$. As expected, the MAE increases both for smaller and larger lengths of the moving average. Importantly, the MAE of both one and two minutes ahead is minimum for a length of the moving average of six minutes. Figure 8.2 shows that the two-minute-ahead MEI is 0.78 when a six minute moving average is used, i.e. 78% of the time the forecast is able to predict the correct tactical decision.
Figures 8.3 and 8.4 show the MAE and the MEI, respectively, as a function of the length of the

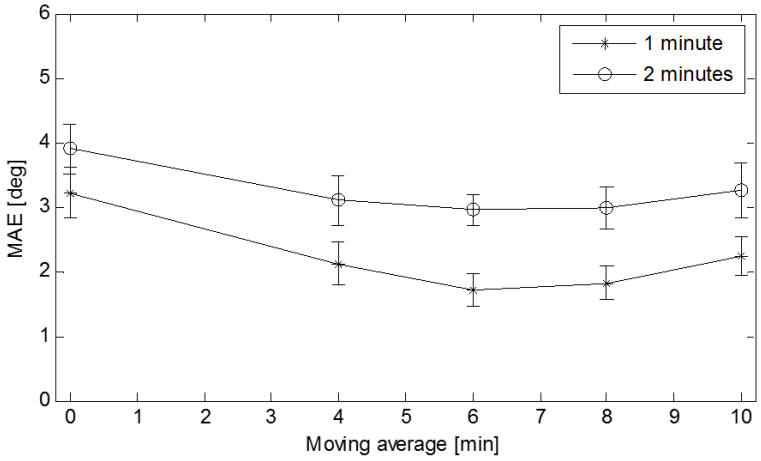

Figure 8.1: MAE versus length of the moving average for wind direction (ANN with 20 hidden neurons and 9 minutes input vector). Figure from Tagliaferri *et al.* (2015).

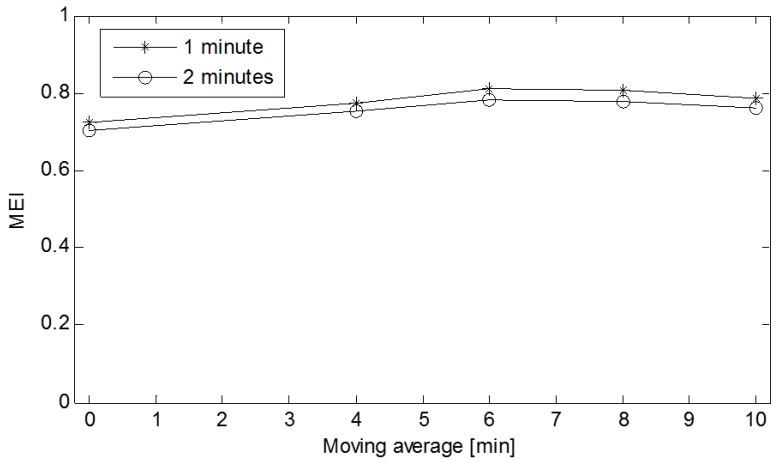

Figure 8.2: MEI versus length of the moving average for wind direction (ANN with 20 hidden neurons and 9 minutes input vector). Figure from Tagliaferri *et al.* (2015).

input vector, where a moving average of six minutes are used to smooth the data set and each hidden layer has 20 neurons. Figure 8.3 shows that the MAE for one and two minutes ahead is a minimum for an input vector size of 18 (9 minutes) and 20 (10 minutes), respectively. The MAE does not show a unique optimum input vector size. However, Fig. 8.4 shows that the two-minute-ahead MEI is a maximum for an input vector size of 18 (9 minutes). Therefore, for this specific application, the best forecast is performed using the past 9 minutes.

Figures 8.5 and 8.6 show the MAE and the MEI, respectively, as a function of the number of

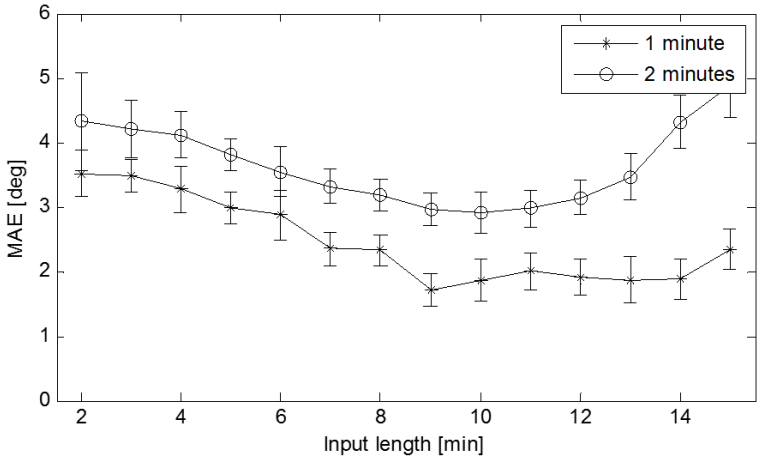

Figure 8.3: MAE versus length of the input vector for wind direction (ANN with 20 hidden neurons and 6 minutes moving average). Figure from Tagliaferri *et al.* (2015).

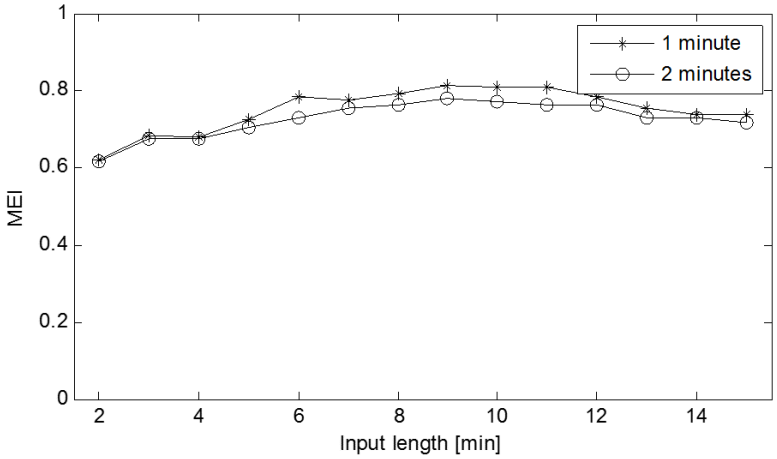

Figure 8.4: MEI versus length of the input vector for wind direction (ANN with 20 hidden neurons and 6 minutes moving average). Figure from Tagliaferri *et al.* (2015).

neurons per each hidden layer, where a moving average of six minutes and an input vector size of 18 (9 minutes) are used. In this case, both indices show a clear optimum when 20 neurons per hidden layer are used.

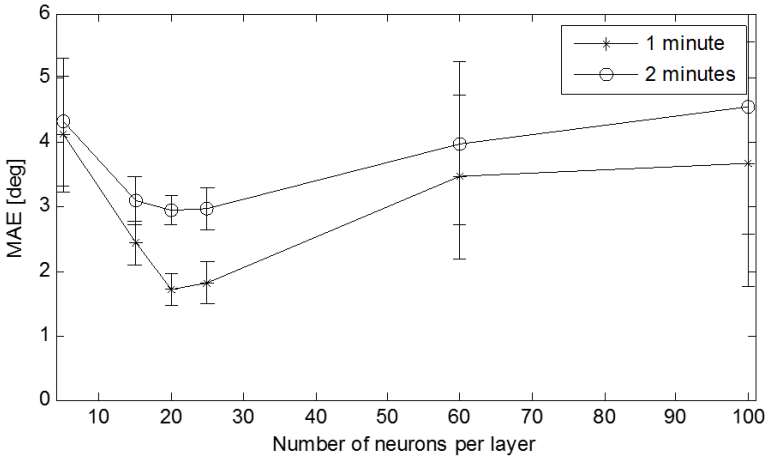

Figure 8.5: MAE versus number of neurons for wind direction (ANN 9 minutes input vector and 6 minutes moving average). Figure from Tagliaferri *et al.* (2015).

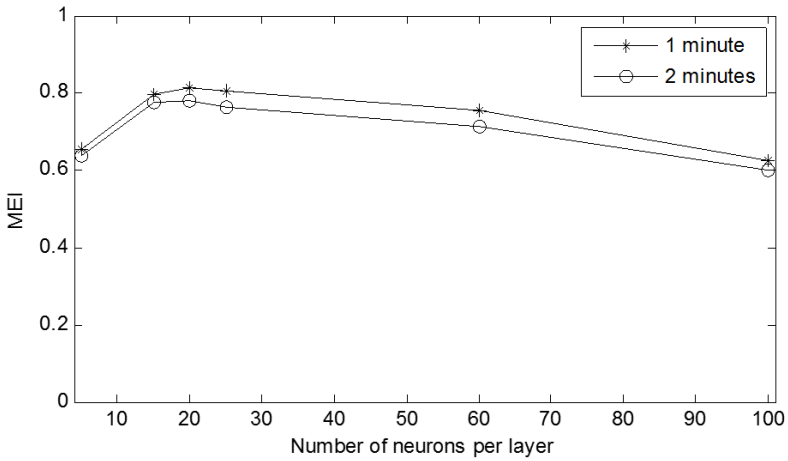

Figure 8.6: MEI versus number of neurons for wind direction (ANN 9 minutes input vector and 6 minutes moving average). Figure from Tagliaferri *et al.* (2015).

8.1.2 Wind speed forecasting

This Section is organised like the previous one, and results relative to wind speed are presented. Figures 8.7-8.12 present MAE and MEI of one and two minutes ahead for different lengths of the moving average, different lengths of the input vector, and different number of neurons in the hidden layers. Again, due to the multidimensionality of the test matrix, the trend of the two

performance indices versus each parameter is shown only for the best combination of the other two parameters. Although wind speed and wind direction have been shown to be not correlated, the optimal ANN configuration for wind speed forecasting results very similar to the one for wind direction forecasting.

Figures 8.7 and 8.8 show the MAE and the MEI as functions of the length of the moving averages, where each hidden layer has 20 neurons and the size of the input vector is 18 (9 minutes). Figure 8.7 shows that for a moving average over 4 minutes, the MAE of one minute ahead is 0.287 ± 0.06 m/s, while the MAE of the two-minute-ahead forecast is 0.382 ± 0.08 m/s. As for the wind direction, the MAE increases both for smaller and larger lengths of the moving average, although this appears to be the parameter that has the most marginal influence on the forecasting ability of the ANN. The MAE of both one and two minutes ahead is minimum for a length of the moving average of four minutes. Figure 8.8 shows that the two-minute-ahead MEI is 0.79 when a six minute moving average is used.

Figures 8.9 and 8.10 show the MAE and the MEI, respectively, as a function of the length

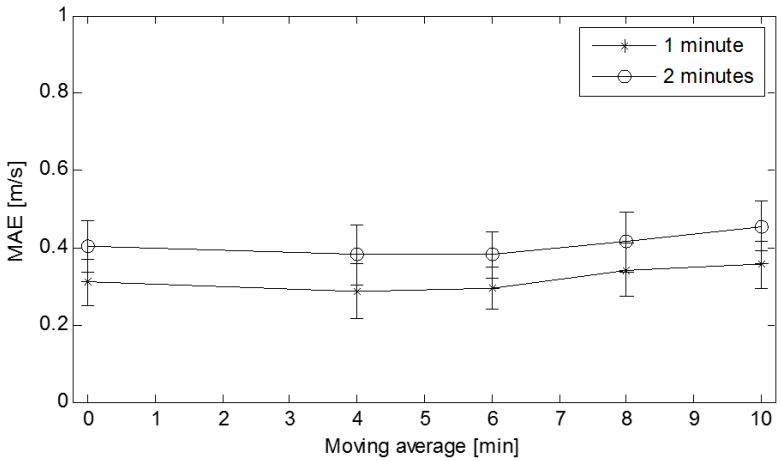

Figure 8.7: MAE versus length of the moving average (ANN with 20 hidden neurons and 9 minutes input vector).

of the input vector, where a moving average of six minutes are used to smooth the data set and each hidden layer has 20 neurons. Figure 8.9 shows that the MAE for one and two minutes ahead does not have an evident minimum, but shows smaller differences between non-optimum setups. However, as for wind direction, Fig. 8.10 shows that the two-minute-ahead MEI is a maximum for an input vector size of 18 (9 minutes). Therefore, for this specific application, the best forecast is performed using the past 9 minutes.

Figures 8.11 and 8.12 show the MAE and the MEI, respectively, as a function of the number of

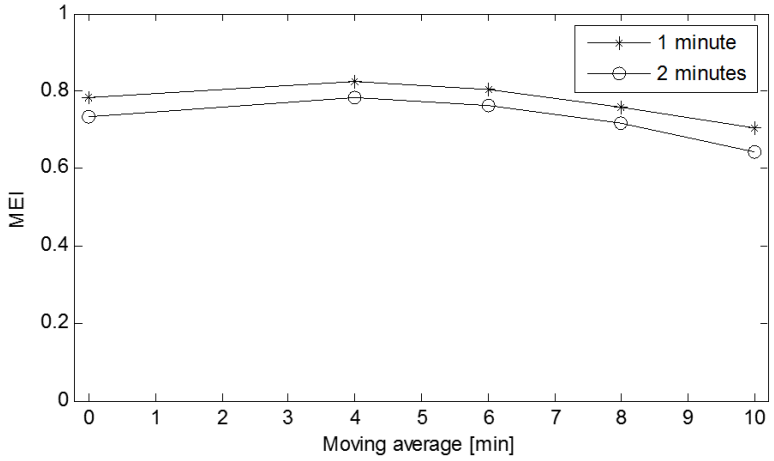

Figure 8.8: MEI versus length of the moving average (ANN with 20 hidden neurons and 9 minutes input vector).

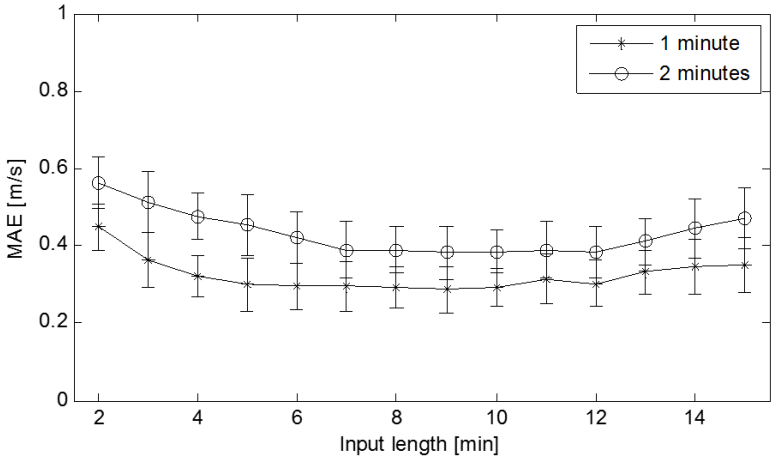

Figure 8.9: MAE versus length of the input vector (ANN with 20 hidden neurons and 4 minutes moving average).

neurons per each hidden layer, where a moving average of six minutes and an input vector size of 18 (9 minutes) are used. In this case, both indices show a clear optimum when 20 neurons per hidden layer are used.

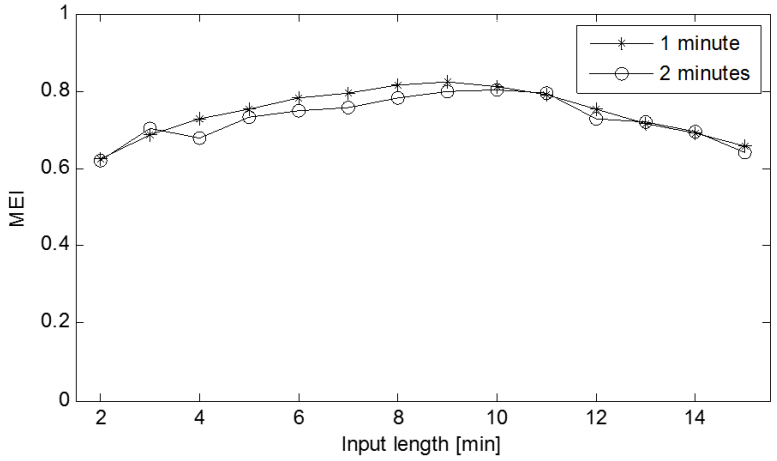

Figure 8.10: MEI versus length of the input vector (ANN with 20 hidden neurons and 4 minutes moving average).

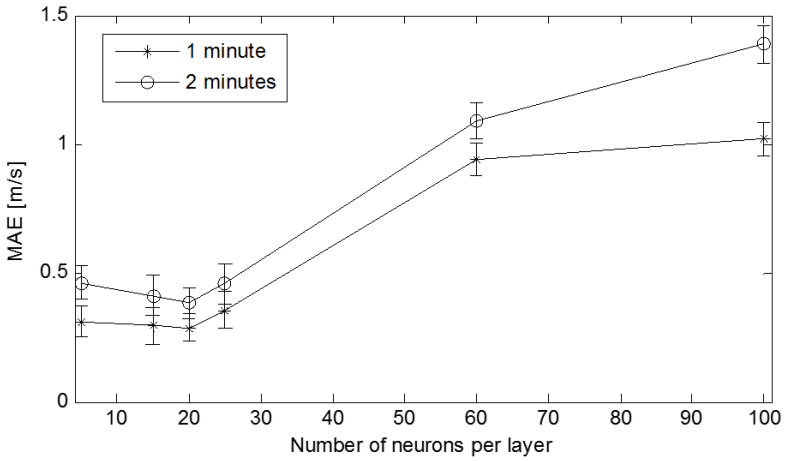

Figure 8.11: MAE versus number of neurons for wind speed (ANN 9 minutes input vector and 4 minutes moving average).

8.2 Optimum ANN model

The module for wind forecasting implemented in the routing algorithm is composed of two ANN ensembles, one dedicated to wind direction and the other to wind speed forecasting. Both models are constituted by ten identical ANN. The ANN for wind direction forecasting are two-layer perceptrons having 20 neurons per layer, taking as input 18 previous values (corresponding to 9 minutes of data) averaged with a six-minutes moving average). The ANN

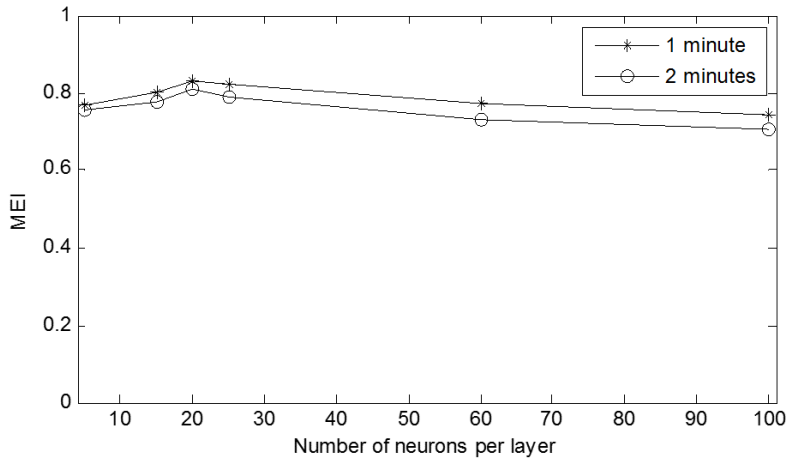

Figure 8.12: MEI versus number of neurons for wind speed (ANN 9 minutes input vector and 4 minutes moving average).

for wind speed forecasting are two-layer perceptrons having 20 neurons per layer, taking as input 18 previous values (corresponding to 9 minutes of data) averaged with a four-minutes moving average).

The performance of the optimised model is summarised in Table 8.1.

Figure 8.13, appeared in Tagliaferri *et al.* (2015) shows an example of wind direction forecast

Table 8.1: Performance of optimised ANN model.

	Wind direction		Wind speed	
	1min	2min	1min	2min
MAE	1.7	3.0	0.287	0.382
MSE	3.7843	8.6121	0.0943	0.2452
R^2	0.9297	0.8189	0.7234	0.6891
MEI	0.81	0.78	0.82	0.79

with a visualisation of the MEI. For clarity, only every second forecast is shown, thus one forecast per minute. For each minute, the solid dots shows the mean recorded wind direction of that minute, while the stars and the circles show the forecast mean wind direction of that minute computed one and two minutes before, respectively. The plots are coloured with vertical bars showing the value of the effectiveness index. Each minute is coloured green (light grey when printed in black and white) if the effectiveness index is one, and is coloured red (dark grey) if zero. Therefore, assuming that tactical choice depend only on the ability of forecasting the need of an immediate tack, for each green minute the forecast would have led to the optimum tactical decision, while for each red minute it could have led to a mistake. In particular, every red minute underlines the cases in which the combination of one-minute-ahead and two-minute-ahead forecasts was not accurate enough to predict the correct tactical decision. In the example

plotted, the ANN forecast leads to the correct tactical choice in all cases but four, corresponding to a MEI of 0.86.

The choice of obtaining the ANN forecast as an ensemble average of the outputs of ten

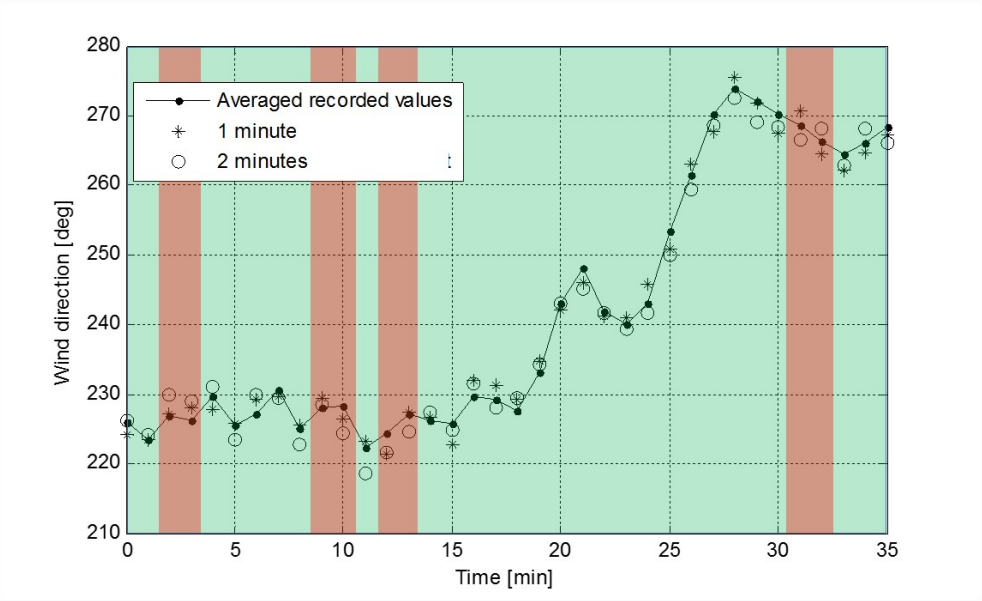

Figure 8.13: Example of forecast for the optimised ANN. Red stripes highlight wrong tactical decision, green stripes correct ones.

networks constitutes a compromise between training time and computational resources. With adequate hardware, and using a less computationally demanding software than Matlab, the variance of the error could be reduced further. As an example, an ensemble average of 1000 networks (instead of 10) was tested for wind direction. Using the test set shown in Fig. 8.13 and the MEI reached 0.97, corresponding to just one potential mistaken decision.

8.3 Verification of the optimum ANN model

In this section some tests on the ANN model are presented in to verify and validate the model. In particular, the following points will be shown

- An AR process can exactly model a time series $x(t)$ for which there exists an integer n and a linear function f such that Equation 8.1 is true

$$x(t) = f(x(t-1), x(t-2), \ldots, f(x-n)) \tag{8.1}$$

- If the above hypotheses are satisfied, the time series can be modelled with equal accuracy by a MLP with just one hidden neuron and linear activation function.
- For a time series $x(t)$ such that

$$x(t) = f(x(t-1), x(t-2), \ldots, f(x-n)) \tag{8.2}$$

The MLP model developed in the previous sections is tested on artificially generated time series. The first time series are constituted by a superposition of sinusoidal waves of different period and amplitude, i.e. they are obtained by sampling a function of the form:

$$x(t) = \sum_{i=1}^{n} a_i \sin(b_i t) \tag{8.3}$$

The variable t is treated as a non-dimensional quantity, as the results shown in this Section do not depend on the unit used; the time step used is equal to 1. An example of a time series is defined in Equation 8.4 and is shown in Fig. 8.14.

$$x(t) = \sum_{i=1}^{5} (2i-1) \sin(2i \cdot t) \tag{8.4}$$

It can be shown that, when coefficients a_i and b_i are integers, a time series of this form can be

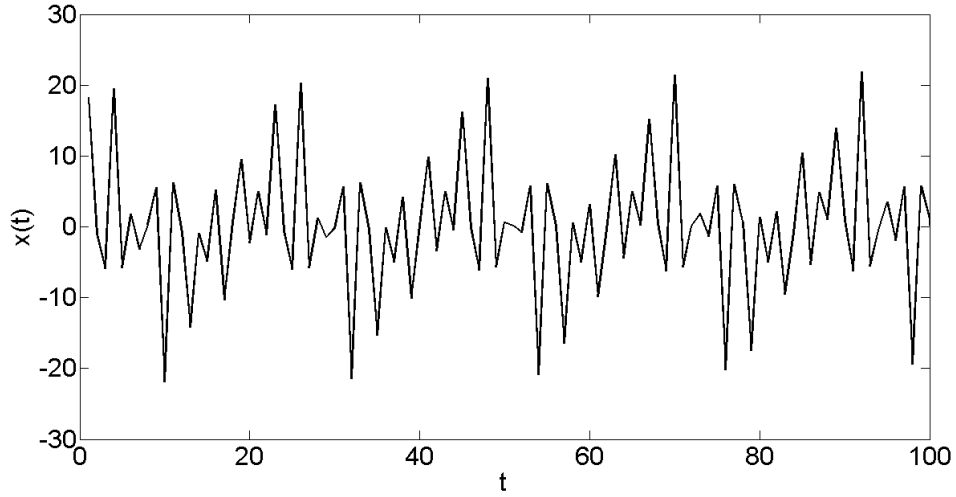

Figure 8.14: Example of artificial time series

exactly modeled with an AR(2n) process. This means that it is possible to find $2n$ coefficients $c_1, \ldots, c_n$ such that

$$x(t) = \sum_{i=1}^{2n} c_i x(t-i) \tag{8.5}$$

and therefore the condition in Equation 8.1 is satisfied. The formal proof of this property is an exercise based on sum formulas for trigonometric functions. Figure 8.15 shows the exact superposition between the series defined in Equation 8.4 and the corresponding AR(10) model. For this case, the AR model is determined by solving the linear system

$$\begin{bmatrix} x(1), & x(2), \ldots & x(10) \\ x(2), & x(3), \ldots & x(11) \\ \ldots & & \\ x(10), & x(11), \ldots & x(19) \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \ldots \\ c_n \end{bmatrix} = \begin{bmatrix} x(11) \\ x(12) \\ \ldots \\ x(20) \end{bmatrix} \tag{8.6}$$

By using a MLP with 10 inputs and one neuron with linear activation function, the plot

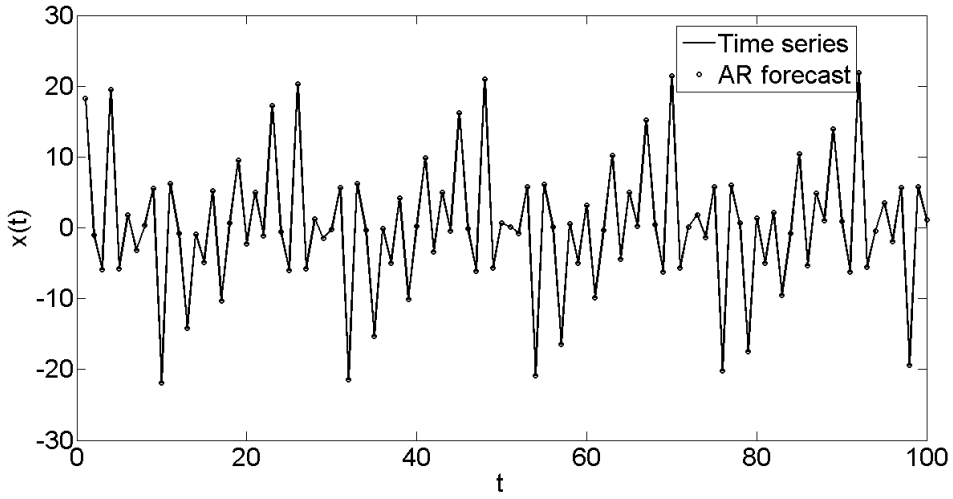

Figure 8.15: Example of AR forecast on artificial time series

in Figure 8.16 is obtained. Figure 8.15 and 8.16 are almost identical, and in both cases the difference between the original time series values $f(t)$ and the approximations computed with the AR model and MLP, respectively, are of the order of 10^{-12}. One limit in the modelling

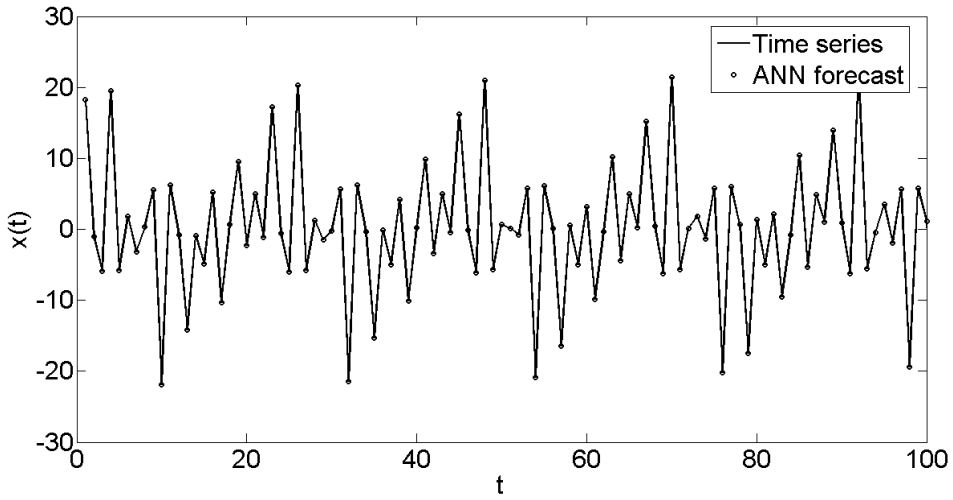

Figure 8.16: Example of ANN forecast on artificial time series

capacity of an AR model arises when not enough input points are used. Using an AR model leads to a systematic error. An ANN model, again having just one neuron, is able to overcome this problem, leading to a better forecast. The difference between the two models can be seen by comparing Figures 8.17 and 8.18, showing, respectively, an AR(9) forecast and an ANN forecast, where both ANN model and AR model take only nine values as input.

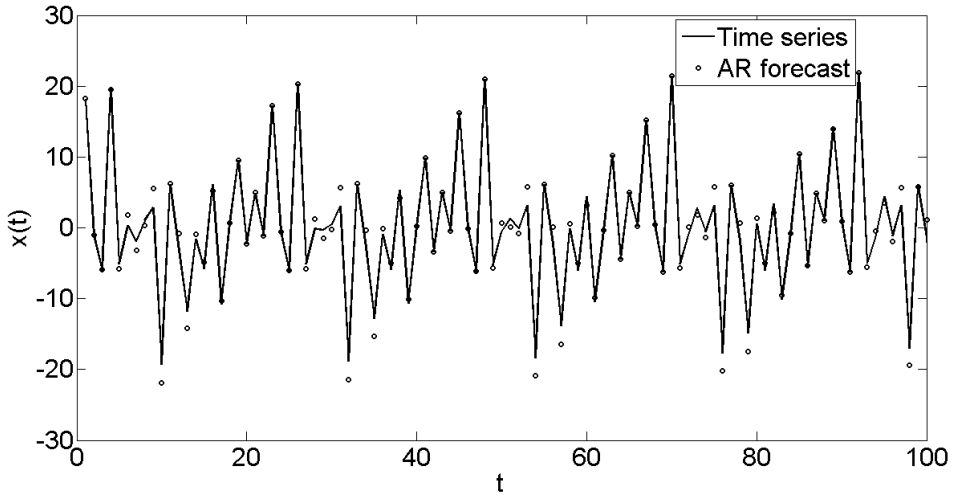

Figure 8.17: Example of AR forecast on artificial time series with suboptimal number of input

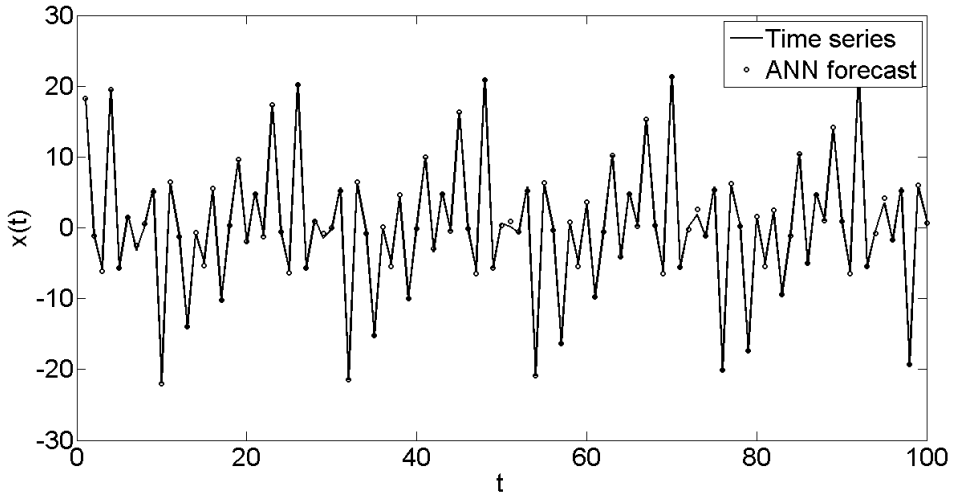

Figure 8.18: Example of AR forecast on artificial time series with suboptimal number of input

8.4 Comparison with other models

8.4.1 Persistence model

Persistence model is the trivial forecasting model based on the assumption that $\mathbf{w}(t+\Delta t) = \mathbf{w}(t)$. Table 8.2 shows the performance indices for the persistence model for wind speed at different time steps. The first row shows the time steps, and the three rows below show the evaluation indices for the persistence model. The dataset presents strings of identical consecutive values, suggesting that there are no significant variations in wind speed on such a small time step. At a time step of 5s, the error is significantly bigger because the variations in wind speed are more significant. An increase of 0.1 m/s in fact may correspond to an increase in boat speed of 0.1 m/s (for an AC72 sailing at $ws = 20$m/s), that over a race leg becomes a time difference of the order of minutes. This underlines the interest in being able to forecast wind

speed variations for a time step of a few seconds.

Table 8.2: Indices for persistence model applied to wind speed

Δt	0.2s	1s	5s	10s	30s	60s	2min	5min
MAE	0.0043	0.0218	0.1065	0.1977	0.3834	0.4898	0.5675	0.6505
MSE	0.0002	0.0014	0.0241	0.0810	0.2837	0.4385	0.574	0.7409
R^2	0.9997	0.9982	0.9720	0.9063	0.6651	0.4762	0.3104	0.1211

Table 8.3 shows the performance indices for the persistence model for wind speed at different time steps. As in the previous table, the first row shows the time steps, and the three rows below show the evaluation indices for the persistence model. As for the wind speed, the error is very small for very short time steps. However there is a significant increase in the error when considering a longer time step.

Table 8.3: Indices for persistence model applied to wind direction

Δt	0.2s	1s	5s	10s	30s	60s	2min	5min
MAE	0.0165	0.0821	0.3994	0.7439	1.6023	2.4998	3.7221	5.9677
MSE	0.0019	0.0238	0.5246	1.7632	6.8246	14.5089	29.5495	73.0405
R^2	0.9999	0.9996	0.9918	0.9728	0.9036	0.8079	0.6599	0.3400

The error comparison between the different models for wind speed and wind direction is shown in Tables 8.4 and 8.5.

Table 8.4: Comparison between different models for wind direction

	Persistence model		ARMA model		SVM model		ANN model	
	1min	2min	1min	2min	1min	2min	1min	2min
MAE	2.499	3.772	2.173	3.432	0.8	1.2	1.7	3.0
MSE	14.5089	29.5495	9.364	14.432	6.854	12.345	3.7843	8.6121
R^2	0.8079	0.6599	0.854	0.775	0.9185	0.8231	0.9297	0.8189

Table 8.5: Comparison between different model for wind speed

	Persistence model		ARMA model		SVM model		ANN model	
	1min	2min	1min	2min	1min	2min	1min	2min
MAE	0.4898	0.5675	0.354	0.426	0.275	0.379	0.287	0.382
MSE	0.4385	0.574	0.135	0.432	0.134	0.284	0.0943	0.2452
R^2	0.4762	0.3104	0.621	0.527	0.714	0.699	0.7234	0.6891

8.4.2 ARMA

Two ARMA models were fitted to wind speed and wind direction time series, respectively, where data were processed in two different ways. In the first case, the dataset was averaged over 30s to obtain the same dataset used for the basic ANN model. In the second case, this dataset was also smoothed by using a moving average over 6 minutes for wind direction and over 4 minutes for wind speed, corresponding to the smoothing used for the optimal ANN models. For both wind speed and direction, the best forecast is obtained when using the dataset

that was not processed with a moving average.

The parameters p and q (defined in Section 5.4.1) are chosen using the BIC criterion. The optimal parameters led to the use of an ARMA(8,8) model for wind direction, and an ARMA(7,6) model for wind speed. The results for wind forecasting with an ARMA model are summarised in Tables 8.4 and 8.5.

8.4.3 Support Vector Machines

In this Section a wind direction forecast is generated using Support Vector Machines. Figures 8.19 and 8.20 show the MAE and MEI, respectively, as a function of the length of the moving average used to pre-process the data, using an input of 8 minutes. In this case the error bars are not needed, as there is only one possible model for each data set. Overall, the SVR shows a significantly lower MAE, both for one and two-minutes ahead predictions. The best performance is achieved for a moving average over 4 minutes, allowing a MAE of 0.8° and 1.2° for one and two minutes ahead, respectively, and a MEI of 0.79. Figures 8.21 and 8.22 show the MAE and

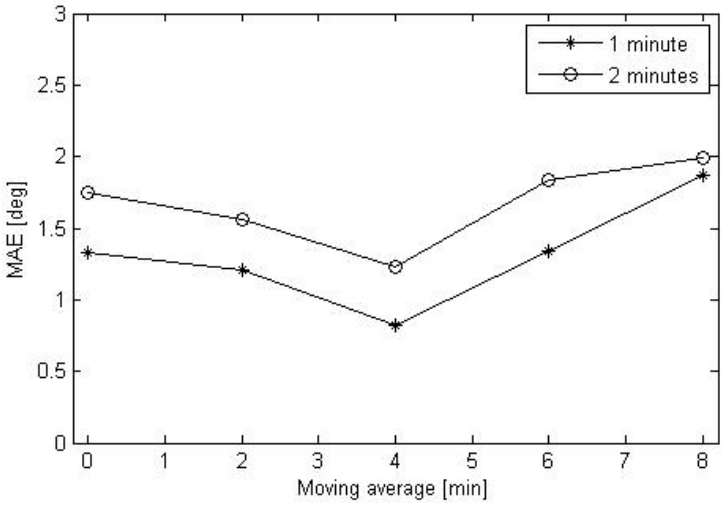

Figure 8.19: MAE versus length of the moving average (SVM).

MEI, respectively, as a function of input vector for the SVR model, where a moving average of 4 minutes is used to smooth the data. The lowest MAE is achieved for an input length of 8 minutes. However, the maximum MEI achievable is of 0.79, differing only by 0.01 from the one achievable with the optimal ANN ensemble forecast. The results for wind forecasting with SVM are summarised in Tables 8.4 and 8.5. While the performance of the optimum ANN and SVM models are similar, it was shown (Section 8.2) that ANN performance can be further improved if a larger ensemble is chosen and if more omputational resources are available. For this reason the ANN model was chosen for this work.

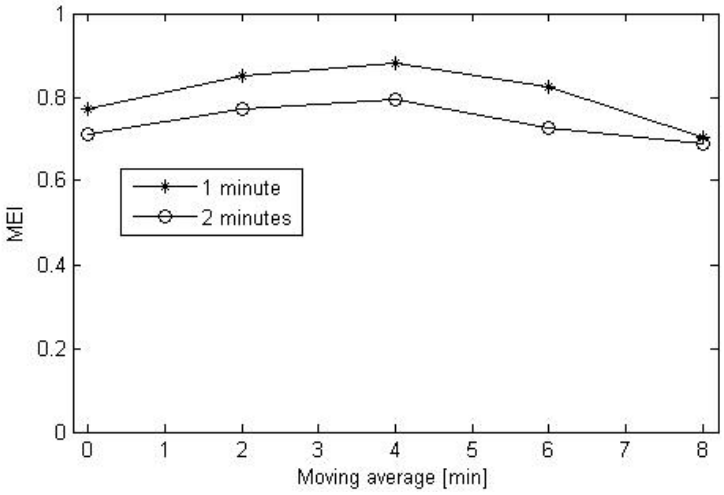

Figure 8.20: MEI versus length of the moving average (SVM).

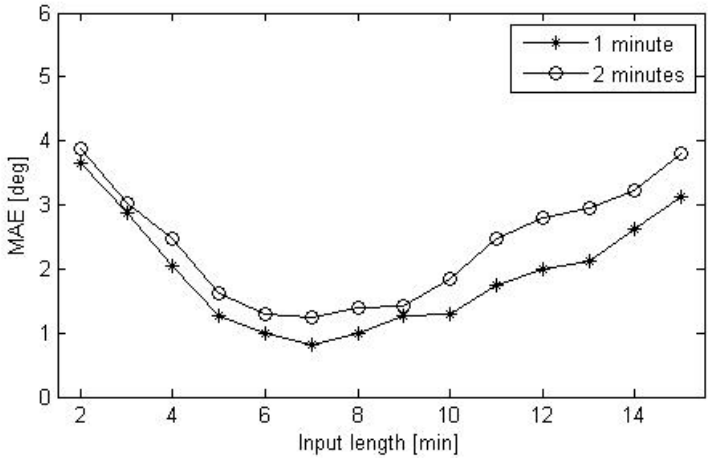

Figure 8.21: MAE versus length of the input vector (SVM).

8.5 Summary

The results included in this Chapter are aimed at demonstrating the suitability of ANN for very short term wind speed and direction forecasting. Two independent models are used for wind speed and wind direction, respectively, and this is motivated by the low correlation that was found in the recorded dataset. A two-layer perceptron was chosen, and the optimal number of neurons per layer was obtained via a set of tests. Similarly, the optimal length of the input vector and the optimal legnth of moving average used to pre-process the input data was optimised.
Despite the low correlation between wind speed and wind direction, the optimal structure found were almost identical. The most crucial parameter appears to be the number of neurons per layer. In fact, if the ANN is too large, the error increases dramatically. For the other parameters, an optimal region rather than an optimal value was found.
The comparison with other forecasting method yields better results for the ANN. The closest competitor is SVR, however, the ANN model allows higher performance with the increase of

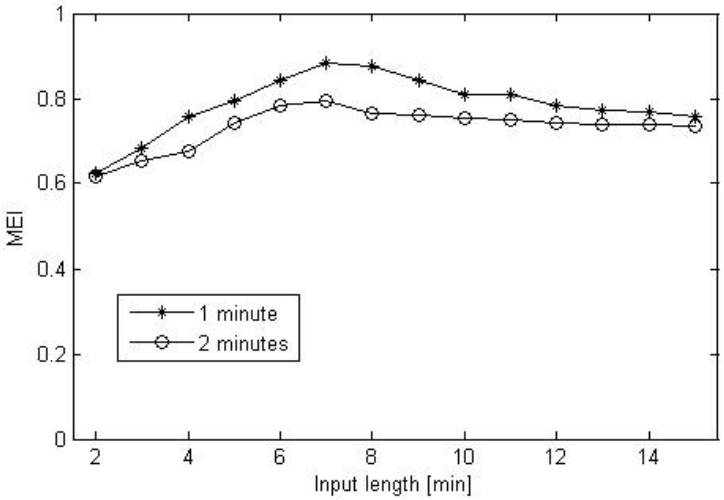

Figure 8.22: MEI versus length of the input vector (SVM).

the available computational resources.

Chapter 9

Dynamic routing for a single boat

This Chapter focuses on the analysis of the strategy generated by the DP algorithm without taking the opponent into account. The first Section is dedicated to the study of strategy in some artificially generated wind scenarios. The first scenarios correspond to simple cases where the wind can be constant, have a consistent shift or some controlled random behaviour. After these cases, some wind scenarios having statistical properties as close as possible to the ones of the dataset recorded in San Francisco are generated and tested. The second Section tests the strategy optimisation for the 13 wind scenarios of the San Francisco data set. The importance of the accuracy of the forecast is tested by comparing strategies computed assuming perfect knowledge of the wind with strategies compared using the ANN forecasting model.

The following cases are selected for testing the algorithm

1. Constant wind speed and direction
2. Constant wind speed and increasing shift towards one side
3. Constant wind direction, increasing wind speed
4. Constant wind speed, temporary shift
5. Recorded wind scenarios

In all cases, the time to tack in the first and last node of the course have been disregarded. Therefore the boat could start and arrive indifferently on the port or starboard tack.

9.1 Constant wind speed and direction

In this Section, speed and direction are assumed constant throughout the race. Figure 9.2 shows the grid generated at step 1. Assuming perfect knowledge of the future behaviour of the wind means that the grid is the same at every time step. The route shown involves the minimal number of tacks (2). Although any other route involving just two tacks would lead to the same completion time, this is the preferred one. In fact, when two different decisions lead to the same time-to-go the one chosen is the one that postpones the tacking manoeuvre.

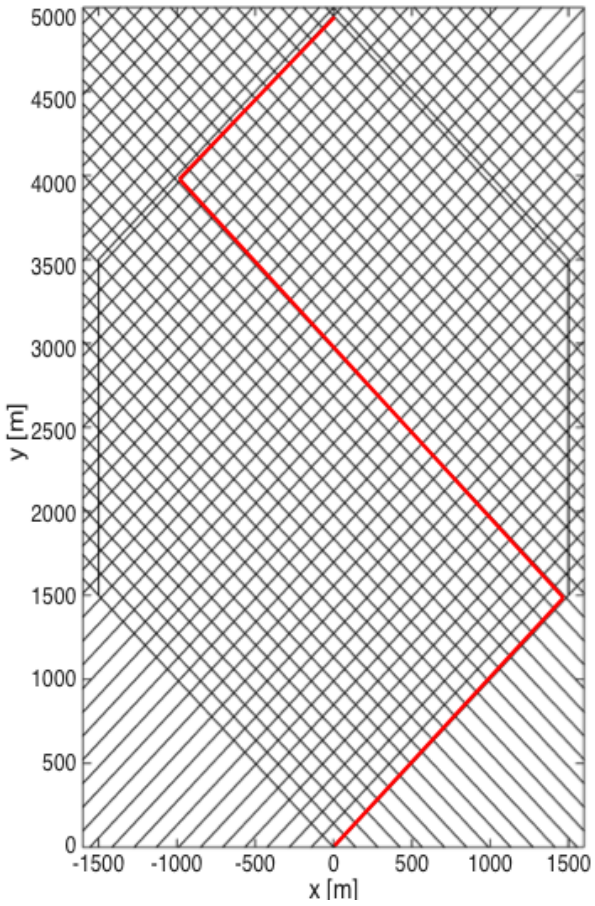

Figure 9.1: Grid generated for constant wind direction and constant wind speed

9.2 Constant wind speed and increasing wind shift towards one side

Figures 9.2 and 9.3 show the grids generated when a constant wind shift towards the left and towards the right, respectively, is forecast. The CWA in the cases plotted goes from 0° to 30° and -30°, respectively. The optimal strategies in this case, as in the previous one, aim at minimising the number of tacks, which is again two. In both cases, sailing in an unfavorable wind is preferred during the first half of the leg in order to get a later advantage of a favourable layline.

Similar cases with different amplitudes of the wind shifts are also tested, and the results of the simulations for a wind speed of 8 m/s are presented in Table 9.1. For this case, the results are all very close. As expected, the optimal number of tacks remains the same, with variations in the time needed to complete the leg of the order of 3%. Those variations are mainly due to the different tacks.

Table 9.1: Simulations for increasing wind shifts assuming perfect knowledge of the wind.

Max shift [deg]	0	5	10	15	20	25	30	35	40	45
Time [s]	345	347	351	349	344	342	339	337	336	335

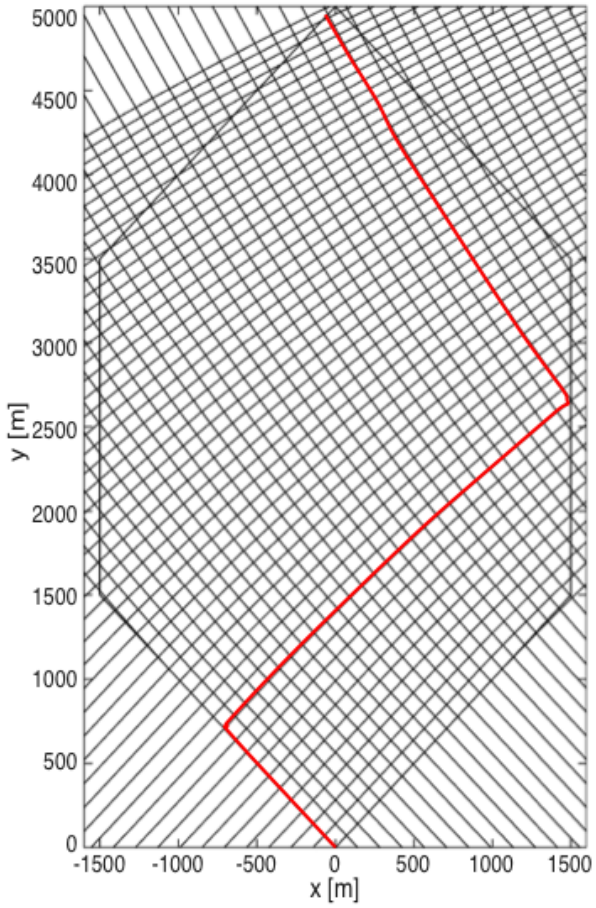

Figure 9.2: Grid generated for consistent wind shift towards the right.

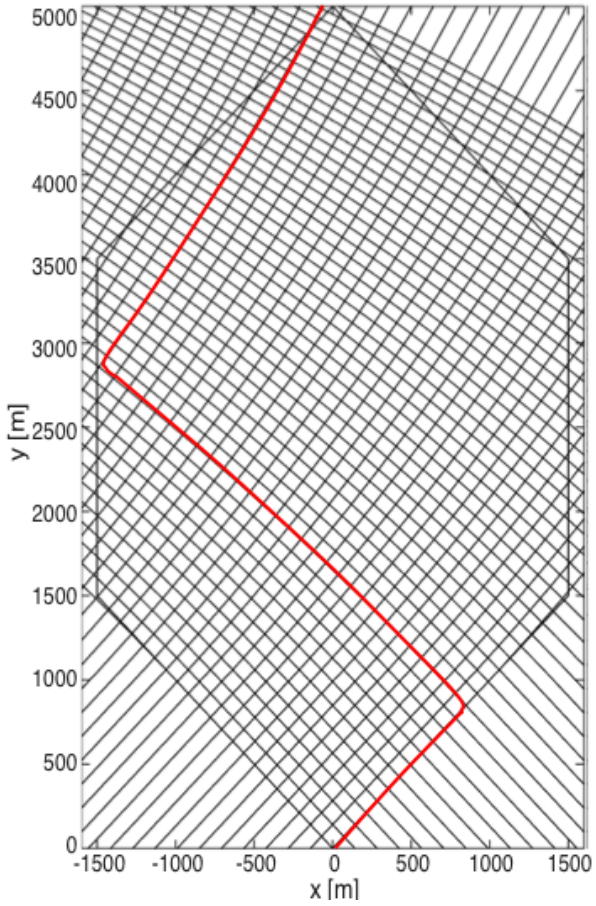

Figure 9.3: Grid generated for consistent wind shift towards the left.

9.3 Constant wind direction, increasing wind speed

For this case, constant wind direction is assumed, with wind speed increasing from 1 m/s to 10 m/s. This has the effect of increasing the distance between nodes of the grid. In fact, the distance between nodes of the grid has the physical meaning of the distance sailed in the time step dt. Figure 9.4 shows the generated grid (changes in the dimension of grid cells are exaggerated in this Figure). The optimal routes followed are similar to the one of Case 1 presented in Figure 9.1. However, the wind-speed dependent tacking penalties mean that the computed strategies give the preference to tacking in a higher wind speed, when the time loss assigned to a tack is less. In general, the changes in wind speed do not affect the strategy as evidently as the changes in wind direction. In the current implementation, the time step is fixed ($dt = 5$s) and therefore

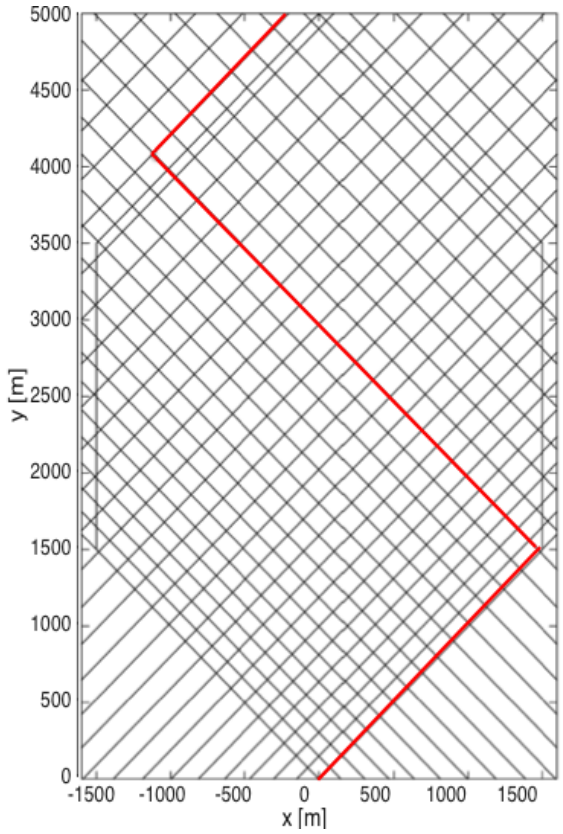

Figure 9.4: Grid generated for increasing wind speed.

the last tack is always after the layline by a time smaller then the time step. Therefore, the higher the wind speed, the larger the maximum path travelled over the layline. Future work can include a variable time step, which decreases when the boat is close to the course boundaries, to the layline, to the other boat etc.

9.4 Constant wind speed, temporary wind shift

When a temporary shift happens, the winning strategy consists in sailing on the tack that allows the optimal VMG. If the beginning and the end of the shift are known, then the optimal strategy simply consists in the choice of the side of the course that allows to be on the right tack. Figure 9.5 shows an example when a wind shift of 30° happens in the first phase of the race.

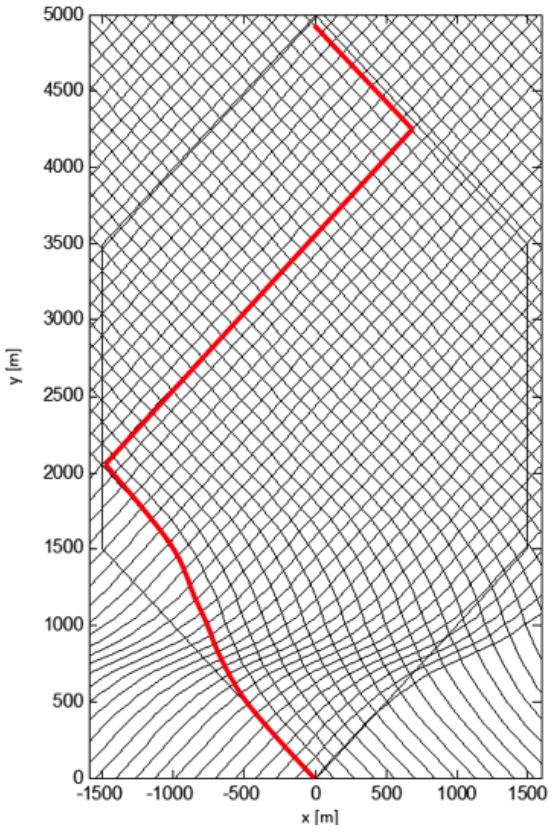

Figure 9.5: Grid generated for consistent wind shift towards the left.

Differently from the previous cases (9.1, 9.2, 9.3), in the case of a temporary shift the wind forecast generated by the ANN model might be different from the exact wind realisation. In fact, for the first three cases, both wind direction and wind speed follow a linear function. Therefore the ANN model is able to forecast the wind behaviour with a negligible error (as shown in Section 8.3) and the resulting computing strategy is the same assuming perfect knowledge of the wind or using the ANN forecast. Conversely, in the present example, the wind direction is constant until a certain point, when it stars increasing linearly up to an arbitrary point, and then it starts decreasing back to the original constant direction. The ANN is able to exactly forecast this behaviour if, and only if, it is trained with identical datasets and the input vector is longer than the wind shift.

9.5 Recorded wind scenarios

The algorithm is tested using the wind data from San Francisco. The performance of the ANN forecast on this dataset has been analysed in Chapter 8.

The results of the simulated races are summarised in Table 9.2. Only 13 upwind legs are simulated using the initial minutes of the last 13 races, as the data relative to the other races was used for training. The average difference between the different times to completion between a

Table 9.2: Simulated races with San Francisco wind dataset.

Race [deg]	1	2	3	4	5	6	7	8	9	10	11	12	13
Time (perfect wind knowledge) [s]	603	743	618	743	642	818	597	661	654	712	748	684	697
Time (ANN forecast) [s]	608	747	634	781	648	821	608	672	659	715	761	693	712
Difference [%]	0.8	0.5	2.5	4.8	0.9	0.4	1.8	1.6	0.7	0.4	0.3	1.2	2.1

boat with a perfect knowledge of the wind and a boat which uses the ANN forecast is 10.7s. A representative example is given in Figure 9.6, corresponding to Race 2. In this example, the difference between the two strategies is limited to a slight delay of the second tack when the ANN forecast is used. One of the worst cases is shown in Figure 9.7, corresponding to Race

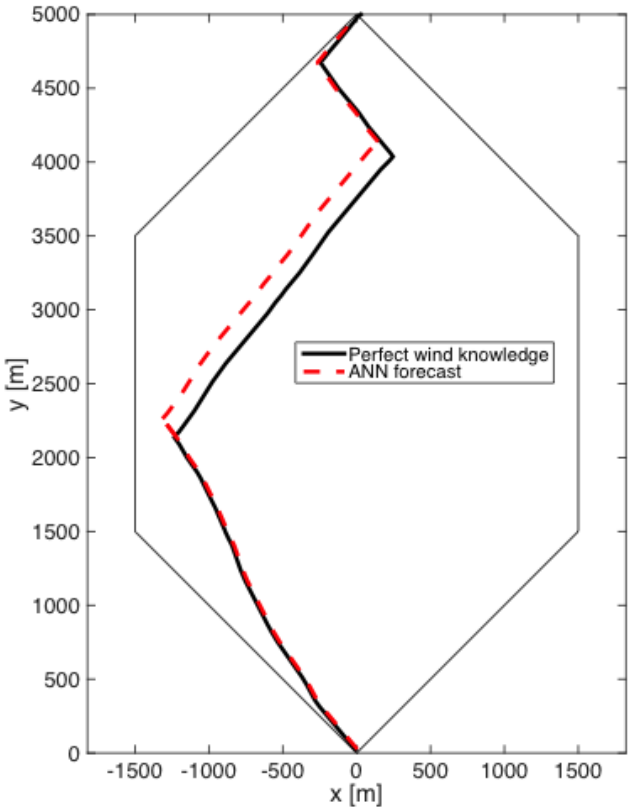

Figure 9.6: Comparison between strategies computed with different wind knowledge - race 2.

4. The black trajectory is the one computed by the algorithm having perfect knowledge of the future wind, while the red dashed one is computed by the algorithm using the ANN forecast. The ANN-based algorithm leads to an extra tack at the beginning of the race, due to a wrong forecast for the end of the race. However, the error is soon recovered and in the final part of the race the two strategies become almost indistinguishable.

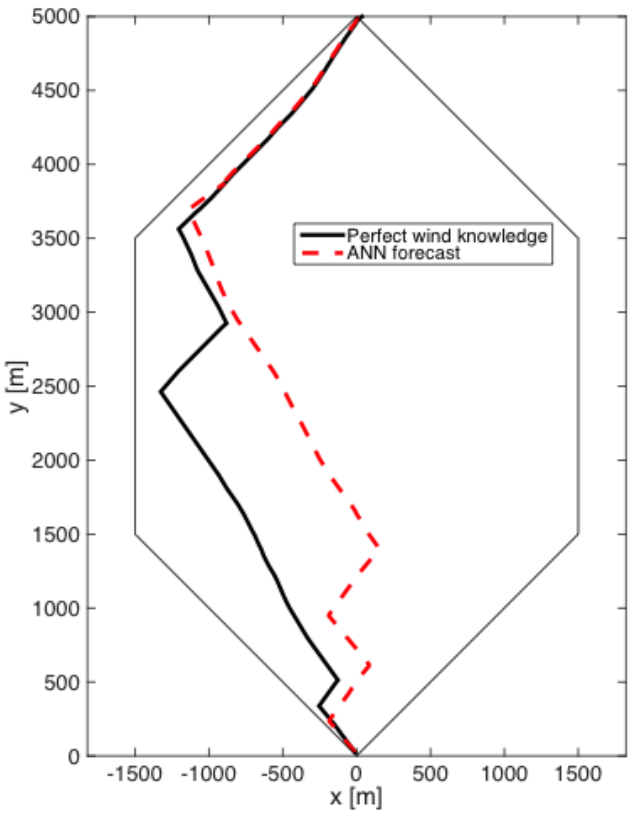

Figure 9.7: Comparison between strategies computed with different wind knowledge - race 4.

9.6 Summary

This Chapter included results of simulated single-yacht races, using both artificially generated and recorded wind time series. The artificial wind was generated to fulfil two aims. Firstly, as a partial verification for the routing algorithm, very simple and controlled wind patterns were generated, including cases with constant speed and/or direction, increasing and temporary shifts. For these cases, the algorithm was able to compute the optimal strategy. The second aim was to be able to simulate a very high number of races, in order to compute the probability distribution of times of arrival for a single yacht following the computed strategy.

The algorithm was shown to lead to the optimal route in all the simple cases, when there is a very evident advantage in choosing one side of the racing area rather than the other, for

instance when there is a constant wind shift. The changes in wind speed are shown to affect the computed strategy less significantly than the changes in wind direction.
The simulations carried out with the recorded wind speed showed that there are minor differences between strategies computed assuming perfect wind knowledge and strategies based on the ANN forecasting model.

Chapter 10

Dynamic routing for two boats

This Chapter presents results of simulated races between two yachts. The yachts are denoted with the letters A and B. The solving algorithm is improved by the addition of a modules for risk modeling. The physical interaction between yachts is modeled according to the procedure presented in Section 6.2.

10.1 Risk management

The importance of risk attitudes comparison is investigated by simulating races where yacht B follows a strategy based on a risk-neutral probability measure for the future wind distribution. The strategy adopted by B is aimed at minimising the time to complete the race leg and corresponds to the procedure described in Chapter 9.
The first set of simulations is based on an arbitrary transformation of the MC transition matrix used for the computation of the expected times. These results are similar to the ones presented in Tagliaferri *et al.* (2014).
As explained in Section 7.4, the original matrix obtained from the dataset from San Francisco is multiplied by a transformation matrix which gives as a result a new transition matrix. With this new distribution, some events are more likely than others, and in particular an ”optimistic” matrix assigns a higher probability to a lifting shift. The matrices used for modelling optimistic and pessimistic attitude and introduced in Section 7.4 are reproduced in Figure 10.1. Depending on the side of the course (left/right), each of the two matrices can correspond to either an optimistic or a pessimistic attitude.
The different attitudes were analysed in single-yacht simulations. Figures 10.2-10.4 present the distribution of the arrival times for yachts following the risk neutral strategy (10.2), an optimistic strategy (10.3) and a pessimistic strategy (10.4). These results confirm that the risk-neutral strategy is the one that leads to the shortest expected time to complete the race. In fact, the risk-neutral strategy is the one which is built to minimise the expected time to complete the race when the probability regulating the wind behaviour is the same that is used to generate the wind scenarios for simulating the race. The distribution presents an asymmetry due to the fact there is a minimum time that a sailor can expect to spend in a race. In fact, even in the best possible scenario the minimum time is constrained by the boat’s polars and the distance to

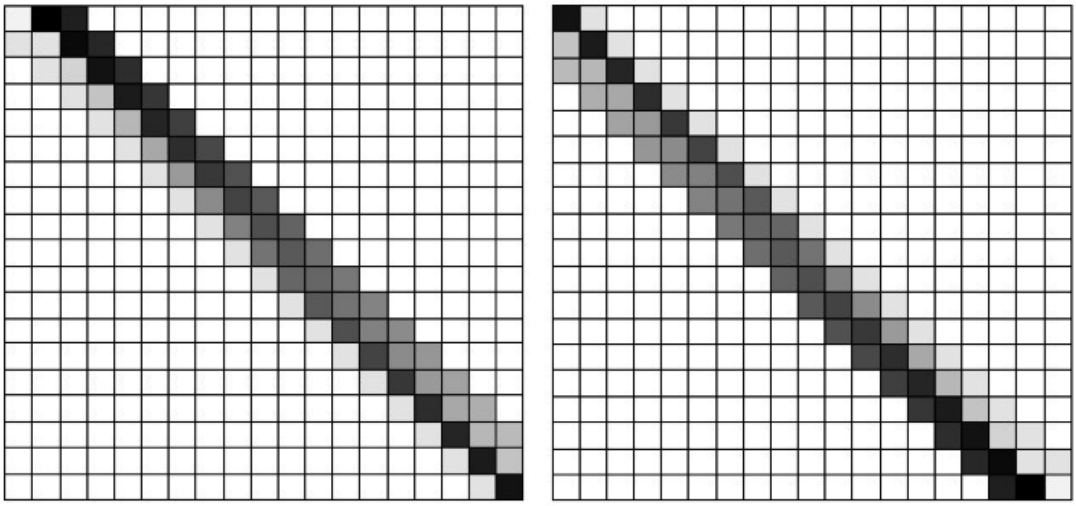

Figure 10.1: Modified transition matrices for a risk-seeking skipper. Advantageous wind shifts occur with higher probability than disadvantageous ones. (Left) Yacht on the left-hand side of competitor and (Right) yacht on the right-hand side of competitor.

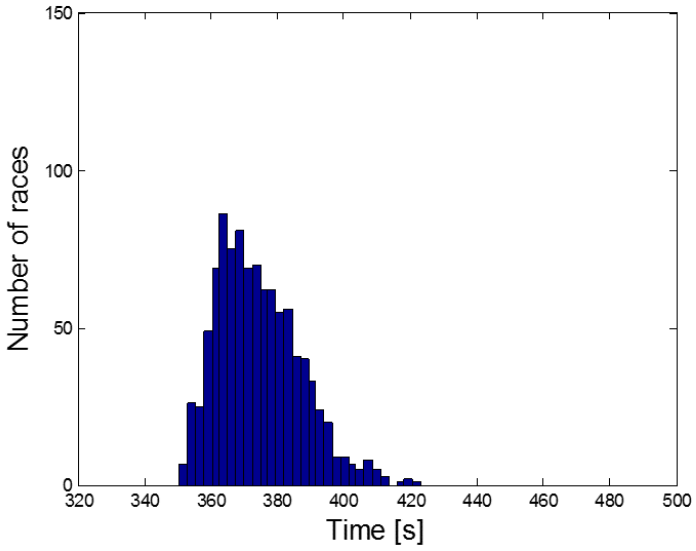

Figure 10.2: Races outcomes for risk-neutral attitude.

race. Conversely, the right tail of the distribution can reach very high values. Although highly unlikely, scenarios with frequent wind shifts of high amplitude and a very low wind speed can lead to virtually arbitrarily long races. If different matrices for computing the expected values

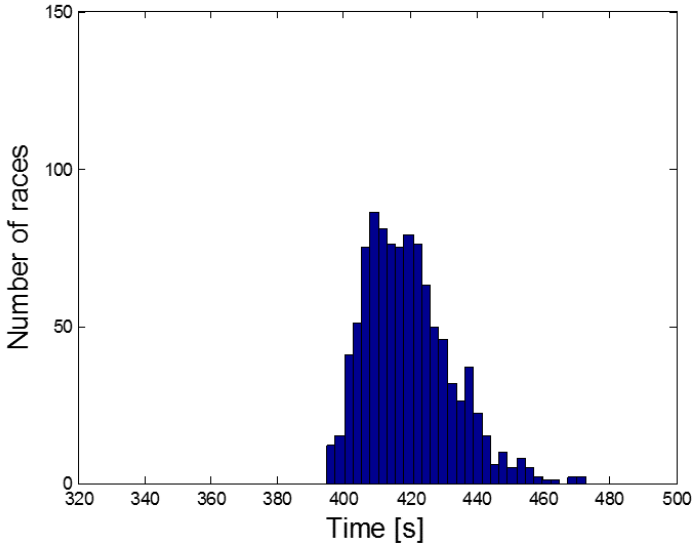

Figure 10.3: Races outcomes for optimistic attitude.

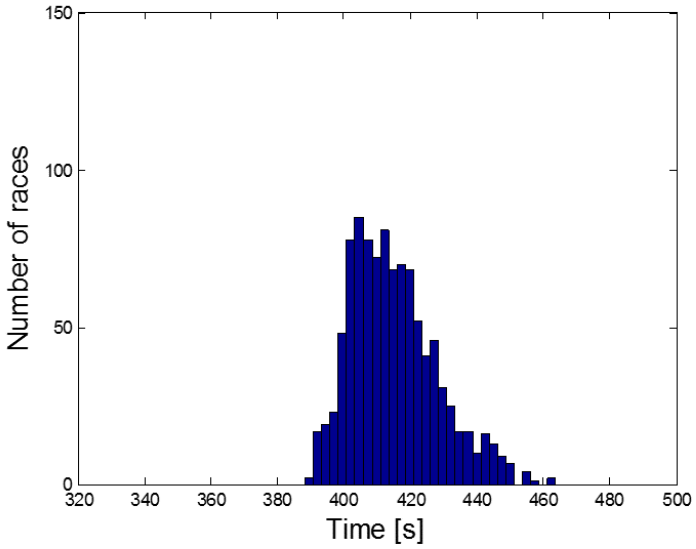

Figure 10.4: Races outcomes for pessimistic attitude.

are used within the DP algorithm, then the average time to complete the race increases.
The optimum risk management is further investigated considering two boats racing each other.

At every step of the simulated race, if A is more than 15s behind B, she uses the risk-seeking, optimistic matrix for the relevant side of the course. If B is more than 15s behind A, she uses the risk-averse, pessimistic matrix. For this case, $T_{switch} = 15$s. The time difference and the matrix transformations are arbitrarily fixed, based on the procedure presented in Section 7.5, and the results obtained confirm the results presented in Tagliaferri *et al.* (2014). Figure 10.5 shows differences between the arrival times of boats A and B. When this time difference is positive, it means that A wins the race. Conversely, if the time difference is negative, B wins the race. This

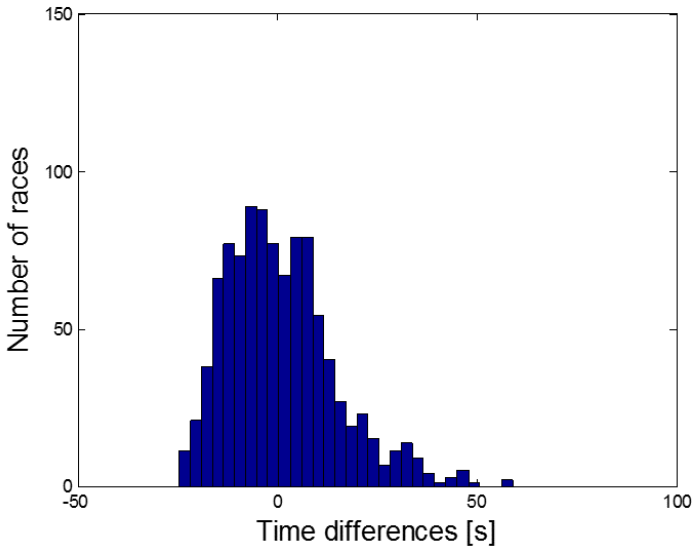

Figure 10.5: Time differences for risk-neutral strategy vs optimistic-pessimistic combination.

set of results confirms that a risk seeking attitude can constitute an advantage for a skipper who is losing the race. However this advantage can be optimised by changing the amount of risk, i.e. how much the new Markov matrices differ from the original one, and the time at which the attitude is changed. i.e. the time difference between the two boats that triggers the attitude switch.

The amount of risk is investigated by comparing strategies obtained using matrices that have been multiplied for the transformation matrix multiple times. The best outcome, is obtained with the use of the matrix shown in Figure 10.6, and by setting $T_{switch} = 10s$. The optimised risk model leads to the distribution in Figure 10.7, which corresponds to a win for boat A in 74%of the cases, and is obtained by post-multiplying the risk-neutral matrix for the square of the original transformation. This optimisation was carried out over a limited set of possibilities, and it must be the subject of further research.

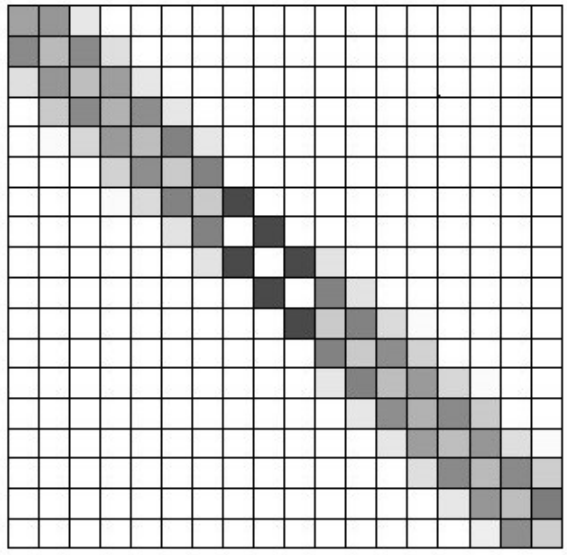

Figure 10.6: Optimal postprocessing matrix

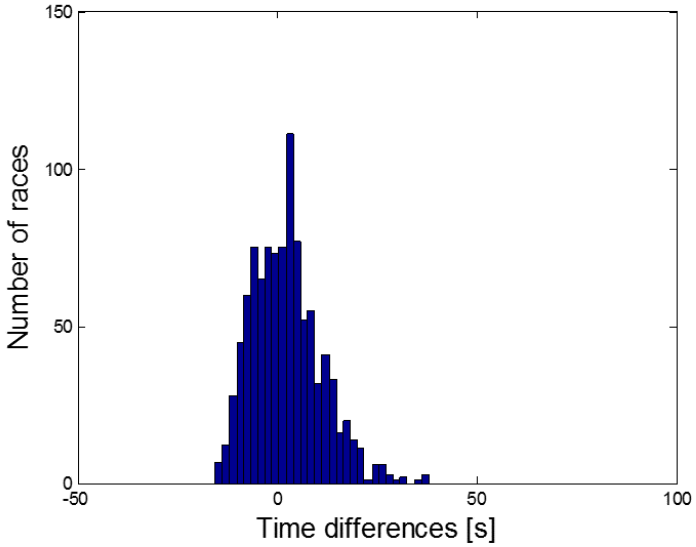

Figure 10.7: Time differences for risk-neutral strategy vs optimal optimistic-pessimistic combination.

10.2 Example of complete race

In this Section a complete race between two boats that follow the optimum course and have an optimum management of risk is presented. The wind which was measured during the last race of the AC is used. Figure 8.13 shows the wind direction and how this is forecast by the two boats. Differently from the original AC race where the wind was recorded, in this example the race starts at minute 18, and is made of only one upwind leg. The wind shows a significant shift towards the right during the race, while the wind speed variations are negligible. Figure 10.8 shows the grid corresponding to the wind realization. The grid shown is coarser than the one computed by the algorithm for clarity.

Both boats A (blue in Figures 10.8-10.13) and B (red in Figures 10.8-10.13) start on a starboard

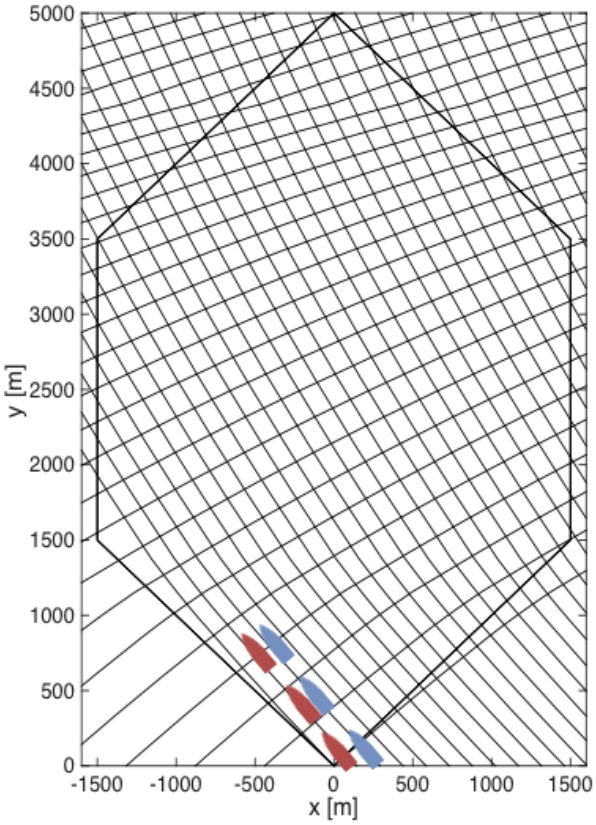

Figure 10.8: Race example - 1

tack (sailing towards the left) and boat B is on the left of boat A, at a distance of 25m (distances between the two boats are magnified in the figures for clarity). Figure 10.8 shows the beginning of the race and the grid represents how the wind was forecast at that time.. Both boats begin the race sailing on a starboard tack. In fact, as a significant increasing shift towards the right is forecast, the best strategy consists in approaching the mark on a starboard tack. Boat A chooses to sail towards the left of the race area up to where, with only one tack, she can reach te right-hand-side layline. This would be the optimal choice for B as well in the absence of A. Unfortunately the more the wind shifts towards the right, the more B finds herself in her area

of unfavourable aerodynamic influence (Figure 10.9). B cannot tack until A tacks because the two boats are too close.

When eventually A tacks (Figure 10.10), B is free to tack as well, but she chooses to wait in

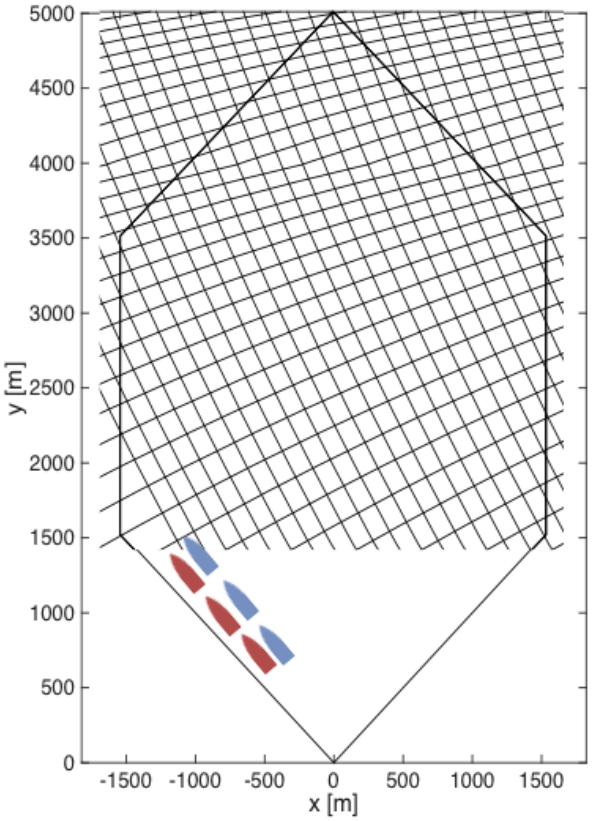

Figure 10.9: Race example - 2

order to perform the tack outside of the area where she would still be slowed down because of the presence of A.

In Figure 10.11 both boats are initially sailing on a port tack, and A is leading. A reaches the layline and tacks to sail towards the mark. B adopts a risk-seeking behaviour, and instead of waiting to reach the layline as well she tacks hoping in a favourable wind shift.

Figure 10.12 shows the end of the race. Although B has managed to avoid A to gain more advantage, she is still slightly behind.

Figure 10.13 shows an alternative realization for this example, where the strategy is computed without taking into account the presence of the opponent. In this case, boat B postpones the second tack until she reaches the racing area right boundary, but this then results again in finding herself in an area of negative influence.

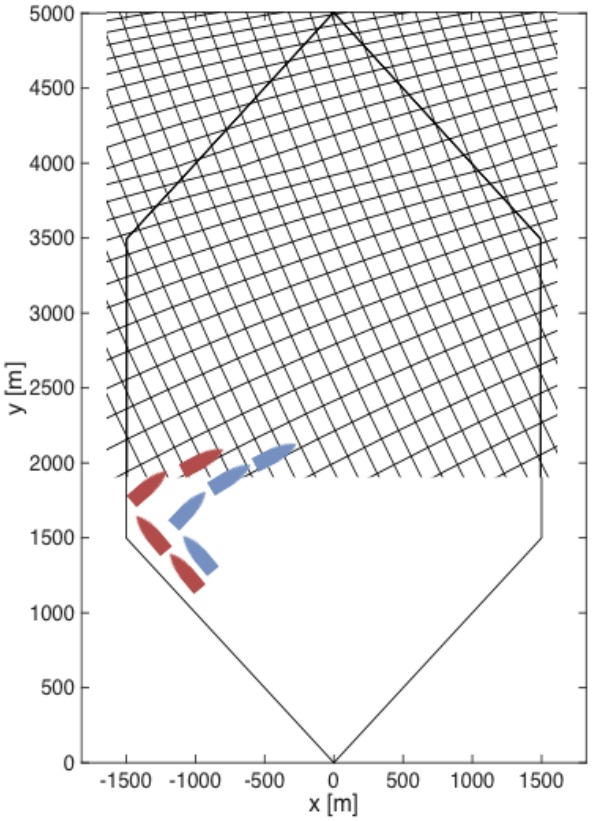

Figure 10.10: Race example - 3

10.3 Summary

Results confirm that the strategy computed with the risk-neutral matrix lead to the lowest expected time when compared to alternative matrices (both corresponding to optimistic and pessimistic matrices). However, when combined, optimistic, risk neutral and pessimistic matrices lead to a win in the majority of cases. This does not correspond to a lower expected time.
Different arbitrary matrices are compared, showing that there is an optimal deviation from the risk neutral matrix that leads to a win in 74% of cases.

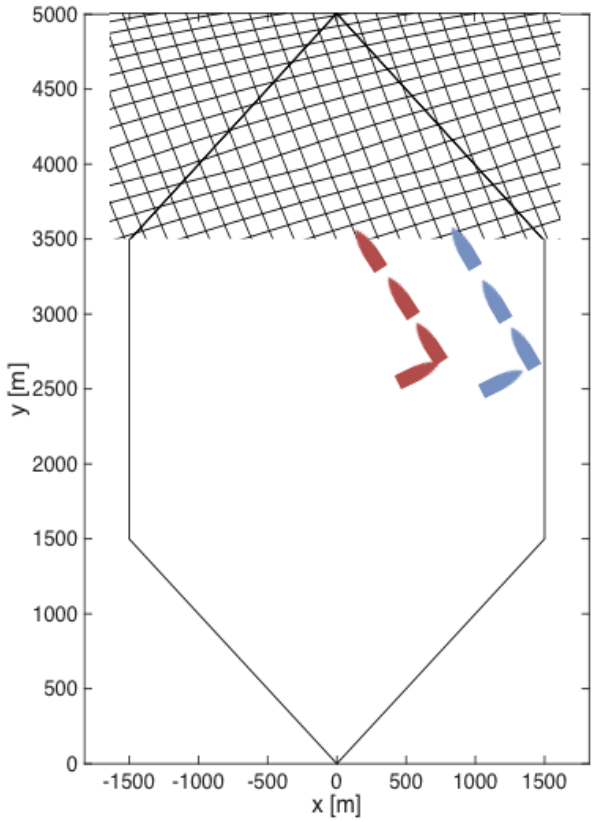

Figure 10.11: Race example - 4

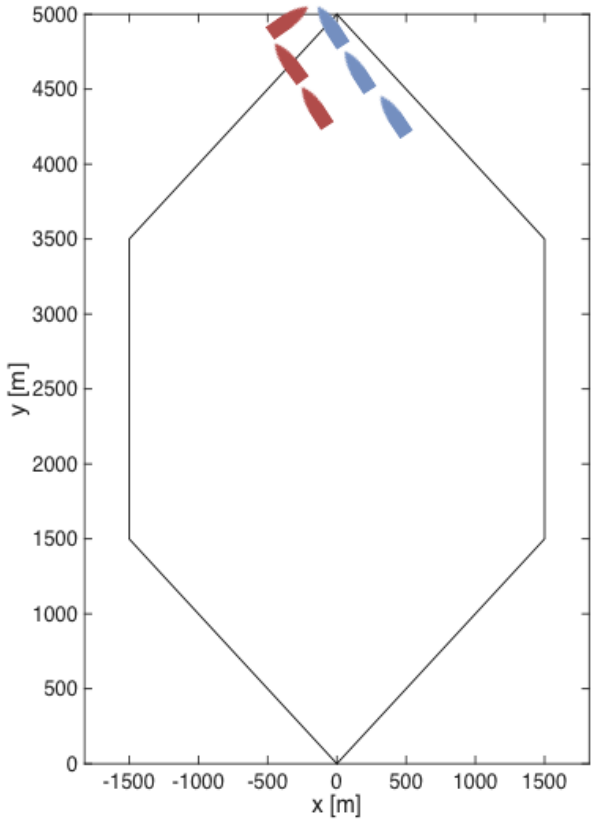

Figure 10.12: Race example - 5

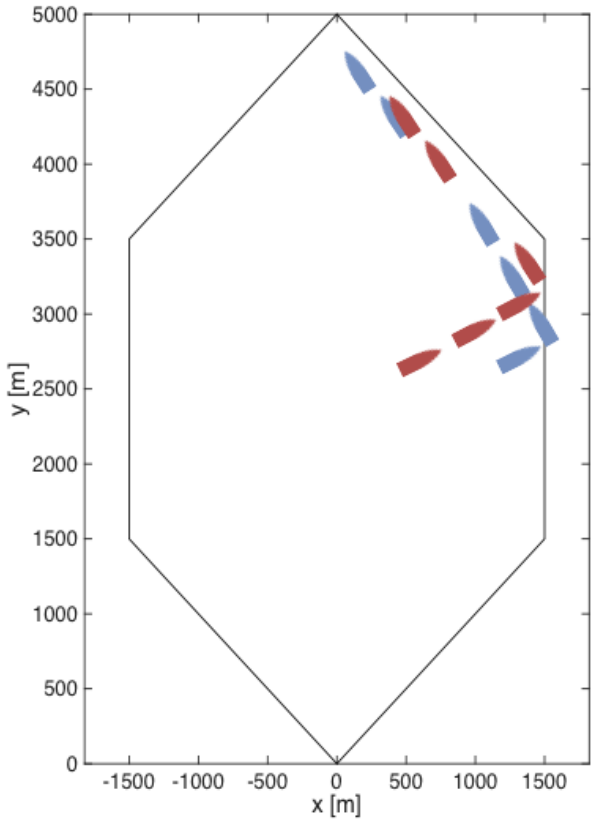

Figure 10.13: Race example - if boats don't see each other

PART IV

Concluding Remarks

Chapter 11

Conclusions

The aim of my work was to investigate the factors influencing the probability of winning a yacht race. I carried out this investigation by developing an algorithm, based on Dynamic Programming, that computes an optimal strategy for a match race. This algorithm combines for the first time Dynamic Programming with a forecasting module based on artificial intelligence, which performs a very-short-term forecast of wind speed and direction, and with a risk model that allows a sailor to tune his risk attitude depending on his position with respect to the competitor.
The main components of the algorithm were first developed and analysed separately, and then joined together in a comprehensive final software capable to compute the optimum course.
This Chapter summarises the major findings presented in this thesis.

- I have developed and implemented an algorithm that, given past wind measurements, boat and opponent's position, is able to compute a strategy that improves the probability of winning the race with respect to strategies aimed at minimising the expected time to complete the race.
- I have used a model based on Artificial Neural Networks for the very-short term forecast of wind speed and wind direction. The parameters defining the model are optimised in order to obtain a forecast that not only has a low absolute error, but that also has a low error when evaluated in terms of tactical decisions taken on the basis of the forecast wind.
- The optimal ANN model leads to improvements of approximately 60% with respect to the persistence model, and performs better than ARMA and Support Vector Machines.
- I have shown that for an upwind leg of the 34^{th} AC, the completion time of a boat which uses the proposed ANN forecast is only 10.7s longer than a boat which has perfect knowledge of the wind.
- I introduced for the first time a risk model based on coherent risk measures in order to investigate whether a change in risk attitude can improve the probabilities of winning a race. An optimistic attitude was associated to a losing skipper, and a pessimistic, conservative one to a winning skipper. Results show that by swapping between those attitudes at different moments of the race can improve the probability of winning up to 64%.
- The risk model was optimised using three different parameters relative to the distance

between boats and the anticipated future wind changes. The results suggest that there is a threshold defining the moment when it is advisable to seek more risk and that not always the risk-seeking and risk-averse behaviours correspond to optimistic and pessimistic anticipations on the wind.

Chapter 12

Future work

The work presented in this thesis can constitute the basis for future research aimed at gaining anew insights on the underlying factors determining a winning sailing strategy. Moreover, the algorithm can be further developed in order to become an effective tool to be used onboard during a sailing race.

- The main improvement that could benefit this work is a validation of the proposed algorithm with full-scale on-water data. This obviously presents some challenges in terms of costs and experiment design.
- The performance of the ANN model for wind forecasting could be further assessed using data sets collected in different locations and, if necessary, its parameters could be refined.
- The wind model could be improved to include spatial variations of wind speed and direction. This would not require major changes in the routing algorithm, as spatial variations can easily be implemented in the grid construction.
- The proposed algorithm could include a tactical model for round marking, which is a critical phase of any race, especially when yachts are close. The challenges associated to mark rounding are of different nature. Firstly, a VPP which can capture the changes in boat speed while rounding a mark be needed. Secondly, the suggested decisions need to compliant the set of racing rules regulating mark rounding.
- The use of coherent risk measures has shown a great potential for risk modeling. However, due to the high dimensionality of the model to be optimised, further investigation needs to be carried out to define the optimal risk-seeking and risk-averse behaviour for a sailor.

Appendix A

VPP definitions and coefficients

The coefficients for residuary resistance of the hull, used in Equation 6.8 are shown in Table A.1

$$\frac{R_{R,hull}}{D_c \rho g} = c_1 + \left(c_2 \frac{L_{CB}}{L_{wl}} + c_3 C_p + c_4 \frac{D_c^{\frac{2}{3}}}{A_w} + c_5 \frac{B_{wl}}{L_{wl}} \right) \frac{D_c^{\frac{1}{3}}}{L_{wl}} + \left(c_6 \frac{D_c^{\frac{2}{3}}}{S_{wc}} + c_7 \frac{L_{CB}}{L_{CF}} + c_8 \left(\frac{L_{CB}}{L_{wl}} \right)^2 + C_g C_p^2 \right) \frac{D_c^{\frac{1}{3}}}{L_{wl}}$$

Table A.1: Coefficients for residuary resistance of bare hull, from Keuning and Sonnenberg (1998).

Fn	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50	0.55	0.60
c_1	-0.0014	0.004	0.0014	0.0027	0.0056	0.032	-0.0064	-0.0171	-0.0201	0.0495	0.0808
c_2	0.0403	-0.1808	-0.1071	0.0463	-0.8005	-0.1011	2.3095	3.4017	7.1576	1.5618	-5.3233
c_3	0.0470	0.1793	0.0637	-0.1263	0.4891	-0.0813	-1.5152	-1.9862	-6.3304	-6.0661	-1.1513
c_4	-0.0227	-0.0004	0.0090	0.0150	0.0269	-0.0382	0.0751	0.3242	0.5829	0.8641	0.9663
c_5	-0.0119	0.0097	0.0153	0.0274	0.0519	0.0320	-0.0858	-0.1450	0.1630	1.1702	1.6084
c_6	0.0061	0.0118	0.0011	-0.0299	-0.0313	-0.1481	-0.5349	-0.8043	-0.3966	1.7610	2.7459
c_7	-0.0086	-0.0055	0.0012	0.0110	0.0292	0.0837	0.1715	0.2952	0.5023	0.9176	0.8491
c_8	-0.0307	0.1721	0.1021	-0.0595	0.7314	0.0223	-2.4550	-3.5284	-7.1579	-2.1191	4.7129
c_9	-0.0553	-0.1728	-0.0648	0.1220	-0.3619	0.1587	1.1865	1.3575	5.2534	5.4281	1.1089

The coefficients for the residuary resistance of the appendages used in Equation 6.9, are shown in Table A.2:

$$\frac{R_{R,app}}{D_{app} \rho g} = c_1 + c_2 \left(\frac{T_{app}}{B_{wl}} \right) + c_3 \frac{T_c + Z_{app}}{D_{app}^{\frac{1}{3}}} + c_4 \frac{D_c}{D_{app}}$$

Table A.2: Coefficients for residuary resistance of appendages, from Keuning and Sonnenberg (1998).

Fn	0.20	0.25	0.30	0.35	0.40	0.45	0.50	0.55	0.60
c_1	-0.00104	-0.00550	-0.01110	-0.00713	-0.03581	-0.00470	0.00553	0.04822	0.01021
c_2	0.00172	0.00597	0.001421	0.02632	0.08649	0.11592	0.07371	0.00660	0.14173
c_3	0.00117	0.00390	0.00069	-0.00232	0.00999	-0.00064	0.05991	0.07048	0.06409
c_4	-0.00008	-0.00009	0.00021	0.00039	0.00017	0.00035	-0.00114	-0.00035	-0.00192

The coefficients for Equation 6.10 are shown in Table A.3

$$\frac{\Delta R}{D_c \rho g} = u_1 + u_2 \frac{L_{wl}}{B_w l} T_c + u_3 \frac{B_{wl}}{T_c} + u_4 \left(\frac{B_{wl}}{T_c} \right)^2 + u_5 \frac{L_{CB}}{L_{wl}} + u_6 L_{CB}^2,$$

Table A.3: Delta resistance hull due to 20°, from Keuning and Sonnenberg (1998).

Fn	0.25	0.30	0.35	0.40	0.45	0.50	0.55
c_1	-0.0268	0.6628	1.6433	-0.8659	-3.2715	-0.1976	1.5873
c_2	-0.0014	-0.0632	-0.2144	-0.0354	0.1372	-0.1480	-0.3749
c_3	-0.0057	-0.0699	-0.1640	0.2226	0.5547	-0.6593	-0.7105
c_4	0.0016	0.0069	0.0199	0.0188	0.0268	0.1862	0.2146
c_5	-0.0070	0.0459	-0.0540	-0.5800	-1.0064	-0.7489	-0.4818
c_6	-0.0017	-0.0004	-0.0268	-0.1133	-0.2026	-0.1648	-0.1174

The coefficients for the computation of the effective span from Equation 6.15 are shown in Table A.4

$$T_{eff} = \left(\left(c_1 \frac{T_c}{T} + c_2 \frac{T_c^2}{T} + c_3 \frac{B_{wl}}{T_c} + c_4 T_{ratio} \right) (c_5 + c_6 F_r) \right) T$$

Table A.4: Coefficients for effective span, from Keuning and Sonnenberg (1998).

h	0	10	20	30
c_1	3.7455	4.4892	3.9592	3.4891
c_2	-3.6246	-4.8454	-3.9804	-2.9577
c_3	0.0589	0.0294	0.0283	0.0250
c_4	-0.0296	-0.0176	-0.0075	-0.0272
c_5	1.2306	1.4231	1.5450	1.4744
c_6	-0.7256	-1.2971	-1.5622	-1.3499

The lift and drag coefficients used in Section 6.1.1 are shown in Tables A.5 and A.6, respectively.

Table A.5: Coefficients for aerodynamic lift, from Larsson *et al.* (2014).

Angle	27	50	80	100	180
Main	1.5	1.5	0.95	0.85	0
Jib	1.5	0.5	0.3	0.3	0

Table A.6: Coefficients for aerodynamic drag, from Larsson *et al.* (2014).

Angle	27	50	80	100	180
Main	0.02	0.15	0.8	1.0	0.9
Jib	0.02	0.25	0.15	0	0

Bibliography

September 2002. URL http://news.bbc.co.uk/sport1/hi/other_sports/sailing/americas_cup/2218797.stm.

URL http://www.tidetech.org/sailing/.

ACEA. 34th ac data specification. Technical report, America's Cup Event Authority, 2013.

Akaike, H. Maximum likelihood identification of gaussian autoregressive moving average models. *Biometrika*, 60(2):255–265, 1973.

Akinci, T. Short term wind speed forecasting with ann in batman, turkey. In *Proceedings of Electronics and Electrical Engineering Conference, OCT*, 2011.

Alexiadis, M., Dokopoulos, P., and Sahsamanoglou, H. Wind speed and power forecasting based on spatial correlation models. *Energy Conversion, IEEE Transactions on*, 14(3):836–842, 1999.

Artzner, P., Delbaen, F., Eber, J.-M., and Heath, D. Coherent measures of risk. *Mathematical finance*, 9(3):203–228, 1999.

Aubin, N. Wind tunnel modeling of aerodynamic interference in a fleet race. Master's thesis, Ecole Centrale de Nantes and Universite de Nantes, 2013.

Banks, J., Webb, A., Spenkuch, T., and Turnock, S. Measurement of dynamic forces experienced by an asymmetric yacht during a gybe, for use within sail simulation software. In *Procedia Engineering - 8th Conference of the International Sports Engineering Association, ISEA; Vienna; Austria*, volume 2, pages 2511–2516, 2010.

Barbounis, T. and Theocharis, J. Locally recurrent neural networks for wind speed prediction using spatial correlation. *Information Sciences*, 177(24):5775 – 5797, 2007.

Barbounis, T. G., Theocharis, J. B., Alexiadis, M. C., and Dokopoulos, P. S. Long-term wind speed and power forecasting using local recurrent neural network models. *Energy Conversion, IEEE Transactions on*, 21(1):273–284, 2006.

Bellman, R. On the theory of dynamic programming. *Proceedings of the National Academy of Sciences of the United States of America*, 38(8):716, 1952.

Bellman, R. E. and Dreyfus, S. E. *Applied dynamic programming*. Rand Corporation, 1962.

Bertsekas, D. P., Bertsekas, D. P., Bertsekas, D. P., and Bertsekas, D. P. *Dynamic programming and optimal control*. Athena Scientific Belmont, MA, 1995.

Bilgili, M., Sahin, B., and Yasar, A. Application of artificial neural networks for the wind speed prediction of target station using reference stations data. *Renewable Energy*, 32(14):2350 – 2360, 2007.

Caponnetto, M. The aerodynamic interference between two boats sailing close-hauled. *International Shipbuilding Progress*, 44(439):241–256, 1995.

Chang, C.-C. and Lin, C.-J. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvmvisitedon01/09/2014`.

Chaouachi, A. and Nagasaka, K. A novel ensemble neural network based short-term wind power generation in a microgrid. *ISESCO Journal of Science and Technology*, 8(14):2–8, 2012.

Chen, B.-J., Chang, M.-W., and Lin, C.-J. Load forecasting using support vector machines: A study on EUNITE Competition 2001. *IEEE Transactions on Power Systems*, 19(4):1821–1830, 2004.

Chen, B., Zhao, L., Wang, X., Lu, J. H., Liu, G. Y., Cao, R. F., and Liu, J. B. Wind speed prediction using ols algorithm based on rbf neural network. In *Power and Energy Engineering Conference, 2009. APPEEC 2009. Asia-Pacific*, pages 1–4. IEEE, 2009.

Du, Y., Lu, J.-p., Li, Q., and Deng, Y. Short-term wind speed forecasting of wind farm based on least square-support vector machine. *Power System Technology*, 32(15):62–66, 2008.

El-Fouly, T., El-Saadany, E., and Salama, M. Grey predictor for wind energy conversion systems output power prediction. *Power Systems, IEEE Transactions on*, 21(3):1450–1452, 2006.

Elman, J. L. Distributed representations, simple recurrent networks, and grammatical structure. *Machine learning*, 7(2-3):195–225, 1991.

Ferguson, D. and Elinas, P. A markov decision process model for strategic decision making in sailboat racing. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6657 LNAI:110–121, 2011.

Fesharaki, M., Shafiabady, N., Fesharaki, M. A., and Ahmadi, S. Using awpso to solve the data scarcity problem in wind speed prediction by artificial neural networks. In *Artificial Intelligence and Computational Intelligence (AICI), 2010 International Conference on*, volume 3, pages 49–52. IEEE, 2010.

Fierro, R. and Lewis, F. Control of a nonholonomic mobile robot using neural networks. *IEEE Transactions on Neural Networks*, 9(4):589–600, 1998.

Focken, U., Lange, M., and Waldi, H. P. Previento – a wind power prediction system with an innovative upscaling algorithm. In *Proceedings of the 2001 european wind energy conference and exhibition*, 2001.

Fonte, P., Silva, G. X., and Quadrado, J. Wind speed prediction using artificial neural networks. *WSEAS Transactions on Systems*, 4(4):379–384, 2005.

Garrett, R. and Wilkie, D. *The symmetry of sailing: the physics of sailing for yachtsmen*. Adlard Coles, 1987.

Gerritsma, J. and Keuning, J. Further experiments with keel-hull combinations. In *SNAME Tampa Bay Sailing Yacht Symposium, St Petersburg USA*, 1985.

Gerritsma, J., Keuning, J., and Versluis, A. Sailing yacht performance in calm water and in waves. In *11th Cesapeake Sailing Yacht Symposium, SNAME*, 1993.

Gerritsma, J., Versluis, A., and Geometry, R. Stability of the delft systematic hull series'. In *International HISWA Symposium on Yacht Design and Construction*, 1981.

Gladstone, B. *Performance Racing Tactics*. North U, 6 edition, 2002.

Gretzky, J. and Marshall, J. The partnership for Americaís cup technology: An overview. In *Proc. 11th Chesapeake Sailing Yacht Symposium*, 1993.

Guigues, V. and Sagastizábal, C. Risk-averse feasible policies for large-scale multistage stochastic linear programs. *Mathematical Programming*, 138(1-2):167–198, 2013.

Guo, Z., Zhao, W., Lu, H., and Wang, J. Multi-step forecasting for wind speed using a modified emd-based artificial neural network model. *Renewable Energy*, 37(1):241–249, 2012.

Hamilton, J. D. *Time series analysis*, volume 2. Princeton university press Princeton, 1994.

Hayashi, M. and Kermanshahi, B. Application of artificial neural network for wind speed prediction and determination of wind power generation output. In *Proceedings of ICEE*, 2001.

Haykin, S. *Neural networks: a comprehensive foundation*. Prentice Hall PTR, 1994.

Hazen, G. S. A model of sail aerodynamics for diverse rig types. In *Proceedings of New England Sailing Yacht Symposium*, 1980.

Hopfield, J. J. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558, 1982.

Hornik, K., Stinchcombe, M., and White, H. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

ISAF - International Sailing Federation. The racing rules of sailing. http://www.sailing.org/documents/racing-rules.php, 2013-2016.

Johnson, M. *Racing basics*. Iowa Sailing, 1995. URL `www.uiowa.edu/~sail/skills/racing_basics/index1.shtml`.

Jung, J. and Broadwater, R. Current status and future advances for wind speed and power forecasting. *Renewable and Sustainable Energy Reviews*, 31:762–777, 2014.

Kerwin, J. and Newman, J. A summary of the h. irving pratt ocean race handicapping project. In *Proceedings of the 4th Chesapeake Sailing Yacht Symposium*, 1979.

Keuning, J. and Binkhorst, B. Appendage resistance of a sailing yacht hull. In *13th Cesapeake Sailing Yacht Symposium*, 1997.

Keuning, J. and Sonnenberg, U. Approximation of the hydrodynamic forces on a sailing yacht based on the'delft systematic yacht hull series'. In *The International HISWA Symposium on Yacht Design and Yacht Construction*. Delft University of Technology, Faculty of Mechanical Engineering and Marine Technology, Ship Hydromechanics Laboratory, 1998.

Keuning, J., Onnink, R., Versluis, A., and Van Gulik, A. The bare hull resistance of the delft systematic yacht hull series. In *International HISWA Symposium on Yacht Design and Construction*, 1996.

Kimball, J. and Story, H. Fermat's principle, Huygens' principle, Hamilton's optics and sailing strategy. *European journal of physics*, 19(1):15, 1998.

Kohonen, T. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480, 1990.

Lahoz, D. and San Miguel, M. Mlp neural network to predict the wind speed and direction at zaragoza. *Monografías del Seminario Matemático García de Galdeano*, 33:293–300, 2006.

Landberg, L. Short-term prediction of the power production from wind farms. *Journal of Wind Engineering and Industrial Aerodynamics*, 80(1-2):207–220, 1999.

Landberg, L. Short-term prediction of local wind conditions. *Journal of Wind Engineering and Industrial Aerodynamics*, 89(3-4):235–245, 2001.

Larsson, L., Eliasson, R., and Orych, M. *Principles of yacht design*. A&C Black, 2014.

Letcher, J. S., Marshall, J. K., Oliver III, J. C., and Salvesen, N. Stars & stripes. *Scientific American*, 257(2):34–40, 1987.

Li, J., Zhang, B., Mao, C., Xie, G., Li, Y., and Lu, J. Wind speed prediction based on the elman recursion neural networks. In *Modelling, Identification and Control (ICMIC), The 2010 International Conference on*, pages 728–732. IEEE, 2010.

Marchaj, C. A., Rusiecki, L., and Barker, S. *Sailing theory and practice*. Adlard Coles, London, 1982.

Marchaj, C. A. *Aero-hydrodynamics of sailing*. Adlard Coles Nautical, London, 1979.

Marti, I., Cabezon, D., Villanueva, J., Sanisidro, M., Loureiro, Y., and Cantero, E. Advanced tools for wind energy prediction in complex terrain. In *Proceedings of the 2003 european wind energy conference and exhibition*, 2003.

Masuyama, Y., Fukasawa, T., and Sasagawa, H. Tacking simulation of sailing yachts–numerical integration of equations of motion and application of neural network technique. In *Proceedings of the 12th Chesapeake Sailing Yacht Symposium*, pages 117–131, 1995.

Milligan, M., Schwartz, M., and Wan, Y. Statistical wind power forecasting models: results for us wind farms. *National Renewable Energy Laboratory, Golden, CO*, 2003.

Mohandes, M. A., Rehman, S., and Halawani, T. O. A neural networks approach for wind speed prediction. *Renewable Energy*, 13(3):345 – 354, 1998.

Monfared, M., Rastegar, H., and Kojabadi, H. M. A new strategy for wind speed forecasting using artificial intelligent methods. *Renewable Energy*, 34(3):845–848, 2009.

More, A. and Deo, M. Forecasting wind with neural networks. *Marine Structures*, 16(1):35 – 49, 2003.

Morgan, N. and Bourlard, H. A. Neural networks for statistical recognition of continuous speech. *Proceedings of the IEEE*, 83(5):742–770, 1995.

Muller, K.-R., Smola, A., Rätsch, G., Schölkopf, B., Kohlmorgen, J., and Vapnik, V. Predicting time series with support vector machines. *Artificial Neural Networks — ICANN'97*, 1327: 999–1004, 1997.

Oossanen, P. V. and Joubert, P. N. The development of the winged keel for twelve-metre yachts. *Journal of Fluid Mechanics*, 173:55–71, 12 1986.

ORC. Vpp documentation. 2013. available online at http://www.orc.org/rules/ORC

Park, S., Lee, J., and Kim, S. Content-based image classification using a neural network. *Pattern Recognition Letters*, 25(3):287–300, 2004.

Parolini, N. and Quarteroni, A. Mathematical models and numerical simulations for the america's cup. *Computer Methods in Applied Mechanics and Engineering*, 194(9):1001–1026, 2005.

Pérez-Llera, C., Fernandez-Baizan, M., Feito, J., and Gonzalez, V. Local short-term prediction of wind speed: a neural network analysis. In *Proceedings of the iEMSs*, volume 2, pages 124–129, 2002.

Philpott, A., De Matos, V., and Finardi, E. On solving multistage stochastic programs with coherent risk measures. *Operations Research*, 61(4):957–970, 2013.

Philpott, A., Henderson, S., and Teirney, D. A simulation model for predicting yacht match race outcomes. *Operations Research*, 52(1):1–16, 2004.

Philpott, A. and Mason, A. Optimising yacht routes under uncertainty. In *The 15th Cheasapeake Sailing Yacht Symposium*, 2001.

Pourmousavi Kani, S. and Ardehali, M. Very short-term wind speed prediction: a new artificial neural network–markov chain model. *Energy Conversion and Management*, 52(1):738–745, 2011.

Renals, S. Radial basis function network for speech pattern classification. *Electronics Letters*, 25(7):437–439, 1989.

Richards, P., Le Pelley, D., Jowett, D., Little, J., and Detlefsen, O. A wind tunnel study of the interaction between two sailing yachts,. *Proc. 21 Chesapeake Sailing Symposium*, 2013.

Scarponi, M., Conti, P., Shenoi, R., and Turnock, S. Including human performance in the dynamic model of a sailing yacht: A matlab®-simulink® based tool. In *RINA - International Conference - Modern Yacht - Papers*, pages 143–156, 2007a.

Scarponi, M., Shenoi, R., Turnock, S., and Conti, P. A combined ship science-behavioural science approach to create a winning yacht-sailor combination. In *18th Chesapeake Sailing Yacht Symposium, CSYS*, pages 1–10, 2007b.

Schölkopf, S. P., Vapnik, V., and Smola, A. Improving the accuracy and speed of support vector machines. *Advances in neural information processing systems*, 9:375–381, 1997.

Schwarz, G. Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464, 1978.

SF Coast Guard, 2013. Dedicated area for the 35th america's cup. URL `http://www.dbw.parks.ca.gov/PressRoom/2013/130703CGAmericasCup.aspx`.

Shapiro, A. and Ruszczynski, A. Lectures on stochastic programming. *Preprint, Georgia Tech*, 2008.

Soman, S., Zareipour, H., Malik, O., and Mandal, P. A review of wind power and wind speed forecasting methods with different time horizons. In *North American Power Symposium 2010, NAPS 2010; Arlington, TX; United States; 26 September 2010-28 September 2010*, 2010.

Spenkuch, T., Turnock, S., Scarponi, M., and Shenoi, A. Lifting line method for modelling covering and blanketing effects for yacht fleet race simulation. In *3rd High Performance Yacht Design Conference 2008, HPYD 2008*, pages 111–120, 2008.

Spenkuch, T., Turnock, S., Scarponi, M., and Shenoi, A. Real time simulation of tacking yachts: How best to counter the advantage of an upwind yacht. In *Procedia Engineering - 8th Conference of the International Sports Engineering Association, ISEA; Vienna; Austria;*, volume 2, pages 3305–3310, 2010.

Spenkuch, T., Turnock, S., Scarponi, M., and Shenoi, R. Modelling multiple yacht sailing interactions between upwind sailing yachts. *Journal of Marine Science and Technology*, 16 (2):115–128, 2011.

Tagliaferri, F. Wind modeling with nested markov chains. *submitted to Wind Energy*, 2015.

Tagliaferri, F., Philpott, A. B., Viola, I. M., and Flay, R. G. J. On risk attitude and optimal yacht racing tactics. *Ocean Engineering*, 90:149 – 154, 2014. Innovation in High Performance Sailing Yachts - {INNOVSAIL}.

Tagliaferri, F., Viola, I. M., and Flay, R. Wind direction forecasting with artificial neural networks and support vector machines. *Ocean Engineering*, 97:65–73, 2015.

Twiname, E. *Start to Win.* Adlard Coles Nautical, Bloomsbury, 1983.

Vapnik, V. *The nature of statistical learning theory*. Springer, 2000.

Viola, I. M. and Flay, R. G. J. Sail pressures from full-scale, wind-tunnel and numerical investigations. *Ocean Engineering*, 38(16):1733–1743, 2011.

Viola, I. M. Recent advances in sailing yacht aerodynamics. *Applied Mechanics Reviews*, 65 (4), 2013.

Welch, R. L., Ruffing, S. M., and Venayagamoorthy, G. K. Comparison of feedforward and feedback neural network architectures for short term wind speed prediction. In *Neural Networks, 2009. IJCNN 2009. International Joint Conference on*, pages 3335–3340. IEEE, 2009.

Whidden, T. *The Art and Science of Sails.* St. Martin Press, New York, 1990.

White, D. J. Utility, probabilistic constraints, mean and variance of discounted rewards in markov decision processes. *Operations-Research-Spektrum*, 9(1):13–22, 1987.

Zheng, H. and Kusiak, A. Prediction of wind farm power ramp rates: A data-mining approach. *Journal of Solar Energy Engineering*, 131(3), 2009.

Ocean Engineering 97 (2015) 65–73

Contents lists available at ScienceDirect

Ocean Engineering

journal homepage: www.elsevier.com/locate/oceaneng

Wind direction forecasting with artificial neural networks and support vector machines

F. Tagliaferri [a,*], I.M. Viola [a], R.G.J. Flay [b]

[a] Institute for Energy Systems, School of Engineering, The University of Edinburgh, United Kingdom
[b] Yacht Research Unit, Department of Mechanical Engineering, The University of Auckland, New Zealand

ARTICLE INFO

Article history:
Received 23 December 2013
Accepted 20 December 2014
Available online 5 February 2015

Keywords:
Wind forecast
Support vector machines
Artificial neural networks
Sailing yacht
Race
Tactics

ABSTRACT

We propose two methods for short term forecasting of wind direction with the aim to provide input for tactic decisions during yacht races. The wind direction measured in the past minutes is used as input and the wind direction for the next two minutes constitutes the output. The two methods are based on artificial neural networks (ANN) and support vector machines (SVM), respectively. For both methods we optimise the length of the moving average that we use to pre-process the input data, the length of the input vector and, for the ANN only, the number of neurons of each layer. The forecast is evaluated by looking at the mean absolute error and at a mean effectiveness index, which assesses the percentage of times that the forecast is accurate enough to predict the correct tactical choice in a sailing yacht race. The ANN forecast based on the ensemble average of ten networks shows a larger mean absolute error and a similar mean effectiveness index than the SVM forecast. However, we showed that the ANN forecast accuracy increases significantly with the size of the ensemble. Therefore increasing the computational power, it can lead to a better forecast.

1. Introduction

The speed of a sailing yacht depends on the wind speed and the course wind angle (the supplementary of the angle between the wind direction and the boat velocity). The boat speed can be represented as a polar diagram, such as the one shown in Fig. 1, where the radial coordinate is the boat speed of an AC72-class boat, for a fixed wind speed, while the angular coordinate is the boat's heading with respect to the true wind direction (data from America's Cup Event Authority, 2013). The dependency of the boat speed on the wind velocity is a key element when deciding the tactics during a yacht race. For instance, when navigating around the world, experienced sailors take advantage of large-scale weather changes, and for inshore races lasting less than one hour, minimal wind shifts can be used to gain an advantage on the competitors. In the latter case, the ability to forecast very-short-term wind changes can make the difference between a win and a loss. A prepared sailor consults wind forecasts before starting the race, and this is usually used to build a strategy aiming at sailing in the racing area where a higher wind speed and a more favourable wind direction are expected. During the race, it is possible to use only the information which is collected on board, including wind speed and direction, which are measured with a cup and vane anemometer on the top of the mast. In the present paper, we present two methods to forecast very-short-term wind shifts based only on the wind direction measured on board during the race. The proposed wind forecast is aimed at complementing the longer-term weather forecast available to the sailors up to the beginning of the race.

1.1. Racing tactics

A typical America's Cup race is made of several turns around an upwind and a downwind mark, where the marks are aligned with the wind. For instance, in an upwind leg, the boats start from the downwind mark and have to reach the upwind mark, located upstream. As shown in the polar plot in Fig. 1, it is not possible to sail directly upwind. The fastest route consists in keeping an optimum course wind angle that maximises the boat velocity in the upwind direction, i.e. the velocity for which the projection on the vertical axis of the plot in Fig. 1 is a maximum. This velocity component in the wind direction is known as VMG. As an example, the optimum course wind angle in an upwind leg for a wind speed of 26 knots is shown in Fig. 1, together with its projection. In most wind conditions this optimum angle is roughly 45°. Because the boat is not sailing straight towards the upwind mark, at a certain

* Corresponding author.
E-mail addresses: f.tagliaferri@ed.ac.uk (F. Tagliaferri), i.m.viola@ed.ac.uk (I.M. Viola), r.flay@auckland.ac.nz (R.G.J. Flay).

http://dx.doi.org/10.1016/j.oceaneng.2014.12.026
0029-8018/

point a change of direction of about 90° will be needed. These changes of course are called tacks. While tacking, the course wind angle decreases from the optimum value, leading to a speed loss. Therefore the total number of tacks should be minimised. Typically, an AC72-class yacht takes about 20 s to complete a tack, and during the manoeuvre the average boat speed is approximately 70% of the optimum speed. Therefore, a tack leads to a loss of about 6 s.

If the wind direction is constant, then the optimum route includes only one tack. Conversely, when the wind direction varies such as in real conditions, then the sailor can tack between two consecutive wind shifts in order to gain an advantage. However, this advantage must be greater than what is lost due to undertaking the tack. There is a high number of possible scenarios that can arise during a race and the problem of taking the best decision is not trivial (Philpott and Mason, 2001). There are however some situations in which the best decision can be easily taken when a sailor is able to foresee how the wind is going to behave. In the following the two most likely scenarios are presented.

Fig. 2 shows the route followed by two boats in order to reach an upwind mark in two different wind conditions. In the first case (left) the wind alternately shifts by 3° to the right and to the left. The first shift is towards the right and, while the two boats sail at their optimum course wind angles, they chose to sail in different directions. Both boats tack at every wind shift. The black boat is always sailing towards the left-hand side of the race course when the wind shift is towards the right, and vice versa when the wind shift is towards the left. Conversely, the grey boat has the opposite strategy. Therefore, the black boat is always sailing to a closer angle to the mark than the grey boat, she sails a shorter course and she arrives first to the mark. The winning tactic of the black boat is that she always maximises her velocity towards the mark. However, this is not always a winning tactic. In fact, in the second case (right), the wind constantly shifts towards the right. The two boats follow the same strategy as in the previous case: experiencing a wind shift to the right, the black boat sails towards the left and the grey boat sails towards the right. Being the wind constantly shifting to the right, the two boats never tack until they reach a lay line, i.e. where a tack allows the mark to be reached without any further tacks. The black boat needs to sail most of the race course before reaching the lay line and being able to tack to the mark, while the grey boat reaches the lay line before the black boat. In this case, even if the two boats have pursued the same strategy based on the wind observed at the time, the resulting course sailed by the black boat is longer than the course sailed by the grey boat, because the wind shifted regularly in the same direction instead of alternating to opposite directions. This shows that the tactical decision cannot be based on the wind direction observed at the time, but that the future wind shifts must be foreseen in order to develop a winning strategy.

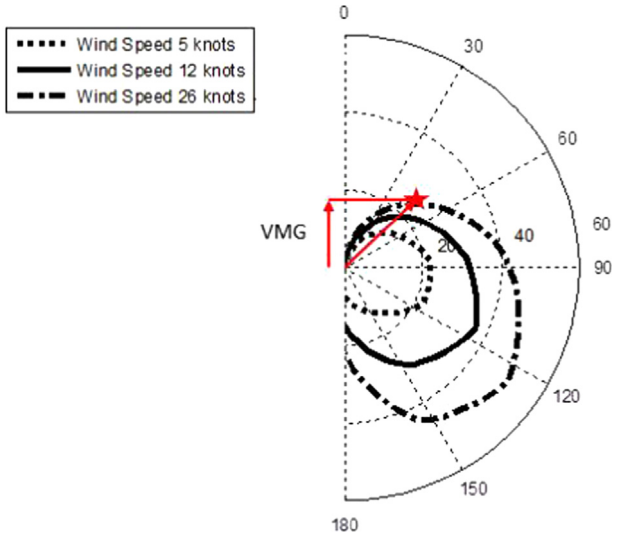

Fig. 1. Polar diagram of an AC72-class yacht.

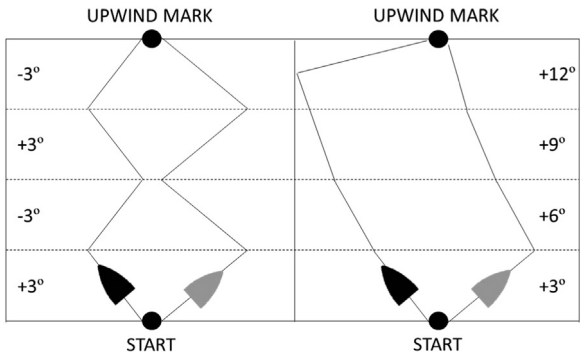

Fig. 2. Yacht routes in oscillating wind (left) and in a permanent wind shift (right).

Fig. 2 shows that the optimum strategy depends on if the wind shift has a wavelength shorter or longer than the distance to the lay line. If the race course is confined by boundaries, such as shore lines or forbidden areas, then wind shifts with wavelength up to the distance to the boundary should be considered.

In the 34th America's Cup, the racecourse was bounded by imaginary lines in order to allow the spectators to be closer to the racing boats. The boats took about two minutes to sail from one boundary to the opposite one, thus the maximum wind shift period of interest was about two minutes.

Consider the black boat in Fig. 2. The boat starts from the centre of the course, she sails for say one minute, then tacks and comes back to the centre of the course. Say she took two minutes plus 6 s for the tack. If the optimum course wind angle is 45°, in two minutes a shift of 3° would lead to an advantage of about 6 s on the grey boat. On the contrary, if the grey boat had tacked at the start and followed the black boat from the beginning, she would be 6 s behind the black boat due to the time spent to tack at the start. Therefore a wind shift of 3° is the threshold at which the decision of tacking or sailing in the same direction has to be made.

1.2. Artificial neural networks

We use artificial neural networks (ANN) to accurately forecast wind shifts larger than 3° for the two minutes ahead. ANNs are computational models that emulate the ability of the human brain to learn from experience, similarly, for instance, to the capability of a human sailor to make predictions based on his lifelong experience. In the literature it is possible to find a vast number of applications where ANNs have been successfully used, especially to solve problems which are peculiar of humans, such as speech recognition (Morgan and Bourlard, 1995), image classification (Park et al., 2004), and control of moving robots (Fierro and Lewis, 1998).

The constitutive unit of a neural network is a neuron, which is a singular processing unit that takes several inputs originating from other neurons, and produces an output that is then transmitted to other neurons. A representation of the structure of a neuron is shown in Fig. 3. A neuron can be broken down into the following components:

1. A set of connecting links, called synapses, where the ith synapse is characterised by a weight w_i (synaptic weight).
2. An adder within the neuron that sums each ith input multiplied by weight w_i.

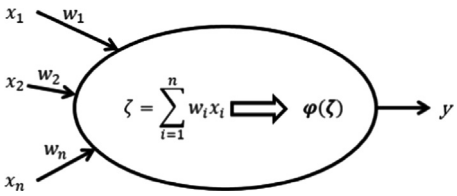

Fig. 3. Schematic representation of a neuron.

3. An activation function, ψ, which transforms the sum computed by the adder into the neuron output y. If the activation function is linear, a neuron results in a linear combination of the input values, while non-linear activation functions allow for the modelling of non-linear problems.

Therefore, a neuron can mathematically be described by the following equation:

$$y = \psi(\zeta); \zeta = \sum_{i=1}^{u} w_i x_i \tag{1}$$

Neurons are assembled together into an integrated structure, the actual ANN, that depends on the kind of problem that the network has to solve. The input vector is processed by an input layer and the information moves through the structure of the network until the output layer.

The learning process, aimed at making the network able to model a specific problem, involves the continuous modification of the synaptic weights. A commonly used training algorithm is based on the principle of iterative error-correction. The synaptic weights of the various neurons are initialised to random values, then a training set of input and output data is presented to the network. For each input vector, the initially generated output vector is compared with the known true output vector. The synaptic weights of the output layer are then modified by adding a factor that is proportional to the error and to a learning rate, and those corrections are extended to all of the weights in the network through a back-propagation process up to the input layer. This operation is iterated until successive changes in the synaptic weights are smaller than a given value, or when the errors begin to increase. For further details on ANNs, including training algorithms and validation processes, see Haykin (1994).

1.3. Support vector machines

In this paper we compare the wind direction forecast performed with ANN with the forecast performed with support vector machines (SVM), which constitute another class of supervised learning models. SVM were originally developed for solving classification problems, where data need to be classified in two or more different categories, and have successively been used also to tackle regression problems (Muller et al., 1997). Regression using SVM is also called in the literature support vector regression (SVR).

The fundamental idea behind SVM is to map the data into a higher dimensional space where the problem is linearly separable, and then solve the problem on the new space. In the case of binary classification this process can be easily visualised as in Fig. 4. In this simple example, points on a plane belonging to two different categories are mapped into a three-dimensional space. The problem is not linearly separable on a plane, but it is linearly separable in three dimensions.

This concept is generalised for SVR, where the aim is to model a function $y = f(\overline{x})$. In this case we have to look for a function Φ which maps the problem to a higher dimensional space where it is

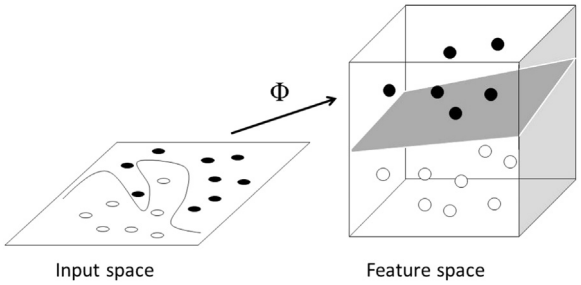

Fig. 4. Mapping of 2D nonlinearly separable data to a 3D space where the problem becomes linearly separable.

possible to perform a linear regression, as shown in the following equation:

$$y = \hat{f}(\overline{x}) = a \cdot \Phi(\overline{x}) + b \text{ where } \Phi : \mathbb{R}^n \to \mathcal{F} \subseteq \mathbb{R}^{n+m}, \quad a, b \in \mathcal{F} \tag{2}$$

A solution to this problem is defined as a minimum for the error function shown in Eq. (8b):

$$R[f] = \sum_{i=1}^{n} C(f(\overline{x}_i) - \hat{f}(\overline{x}_i)) + \lambda |b| \tag{3}$$

where n is the sample size, C is a cost function and the term $\lambda|b|$ is added to enforce flatness in the higher-dimension space. The function Φ can be found as the unique solution of a quadratic programming problem (Muller et al., 1997; Vapnik, 2000). Eq. (2) can be rewritten in terms of this solution as

$$\hat{f}(\overline{x}) = \sum_{i=1}^{n} (\alpha_i - \alpha_i^*) k(\overline{x}_i, \overline{x}) + b \tag{4}$$

where k is a symmetric kernel function (see Schölkopf et al., 1997 for details) and α_i, α_i^* constitute the solution to the quadratic programming problem. Details of this method can be found in Muller et al. (1997).

An important difference between SVR and ANN approaches is that SVR lead to a unique deterministic model for each data set, while ANNs depend on a random initial choice of synaptic weights. Therefore, in order to minimise the effects of this randomness in an ANN model, we train different ANNs and then average their outputs. This is not necessary for SVR, which constitutes an advantage in terms of computational time.

1.4. Wind speed and wind direction

The vast majority of the literature is focused on wind speed forecasting, as opposite to wind direction forecasting (Bitner-Gregersen et al., 2014; Costa et al., 2008). ANNs have been used for wind speed forecasting, ranging from hours (Li and Shi, 2010) to days (Barbounis et al., 2006; Fierro and Lewis, 1998), and have been mostly motivated by application in renewable energy. The wind direction has been used as an input for wind speed forecasting together with other quantities such as humidity and pressure to improve the forecast of the wind speed (Mabel and Fernandez, 2008), which is the key parameter to predict the amount of energy available in the wind stream. Conversely, in the present application we are focused on nowcasting wind direction, i.e. on wind direction forecasting of few minutes ahead. In fact, an increase in the wind speed would lead to an increase in the boat speed, but the wind speed distribution over the racecourse is forecast more easily than the wind direction by visual observation of the surface of the sea. On the contrary, wind direction is forecast with difficulty by visual observation. Using wind speed, wind direction and other time series together can improve the accuracy of forecasts (Mabel and Fernandez, 2008),

but for nowcasting, wind speed and direction may be uncorrelated and therefore should not be used together. In particular, as discussed in the next section, the input data set that we used was particularly uncorrelated and using the wind speed as additional input would have not improved the forecast of the wind direction.

2. Method

2.1. Input and output data

The data set used for this work consists of registrations collected during the 34th America's Cup in San Francisco (America's Cup Event Authority, 2013). Wind speed and direction are recorded from different moving and fixed sources at a frequency of 5 Hz, and averaged by the Media Data Server System with a proprietary algorithm.

Because the data is the average of several measurements taken in different locations across the race course, records for the same day show a correlation coefficient spanning from −0.4 to 0.6, while the global correlation coefficient was −0.38. Also the correlation coefficient is highly volatile with respect to the subset used. Therefore only the wind direction is used as input. As far as known by the present authors, this is the first wind direction forecast based on ANN or SVR using only wind direction as input.

The data include registrations collected during 34 days. We use the last 100 min to test the performance of the ANN and SVR forecasts, and the rest of the data to train both models. Due to the high sampling frequency (5 Hz) and the limit of precision to 0.1 degrees, several consecutive values of the data set were identical. Therefore we average the data over 30 s using 150 consecutive values, leading to a re-sampled dataset with one value every 30 s (0.03 Hz).

As an example, the last 35 min of the re-sampled data set, corresponding to the last race of the 34th America's Cup finals, are shown in Fig. 5. The beginning of the race is characterised by small wind shifts, such as those on the left of Fig. 2, while in the second half of the race there is a consistent wind shift towards the right such as on the right of Fig. 2. During the 35 min period shown, the wind shifts by more than 3° for periods in excess of 30 s 16 times.

The time history of the re-sampled data set shows significant fluctuations with continuous change of direction every 30 s. In order to further smooth the data set, a moving average is used. The length of the moving average is optimised based on the performance indices, as discussed in Section 3. Increasing the length of the moving average, the time series results increasingly smoother, while the sampling frequency remains constant. As an example, in Fig. 5 the data set smoothed with a moving average of six minutes is also shown.

A vector of consecutive past data was used as input and its length was optimised over the indexes described in Section 2.4.

The outputs of the models are the wind directions averaged over one minute ahead, and averaged between one and two minutes ahead. Therefore we use an ANN and SVR to approximate a function f that expresses future values as a function of past ones as in the following equation:

$$\left(\frac{x_{t+1}+x_{t+2}}{2}, \frac{x_{t+3}+x_{t+4}}{2}\right) = f(x_t, x_{t-1}, \ldots, x_{t-m}) \quad (5)$$

where x_t is the wind speed at discrete time step t and m is an optimised integer.

2.2. ANN forecast

The chosen structure for the ANN is a feed-forward multi-layer perceptron which was implemented in Matlab. The hyperbolic tangent sigmoid function was used as activation function. Fig. 6 shows one of the feed-forward structures tested. In a multi-layer structure, neurons are organised in layers, where neurons do not receive input from any other neuron in the same layer. In a feed-forward structure, the information flows in only one direction, i.e. the output of each layer is input for the successive layer. Both the number of layers and the number of neurons per layer can be increased at discretion, but in general any network with more than two layers can be reduced to a two-layers network with an adequate number of neurons (Haykin, 1994). The example shown in Fig. 6 shows a network with two hidden layers of five neurons each, taking past wind directions as input and giving as output the average wind direction in the following two minutes. In this study we present a comparison between two-layer perceptrons with different number of neurons. In Section 3 we present a parameter study for different lengths of the input vector, lengths of moving average and number of neurons. It will be shown that there is an optimum for each of these parameters that allows the maximum accuracy of the forecast.

Because the synaptic weights are randomly set and then iteratively adjusted with the training of the network, networks initialised with different sets of random weights lead to slightly different output. Therefore what it is called *ANN forecast* in the following is the average of the outputs of one ensemble made of ten identical networks trained independently using different sets of initial synaptic weights. This averaging process decreases the chances of large errors due to unfortunate initialisation-training combinations (Chaouachi and Nagasaka, 2012). The size of the ensemble is chosen arbitrarily and in Section 3 it will be shown that a larger ensemble can lead to a more accurate forecast but to higher computational time.

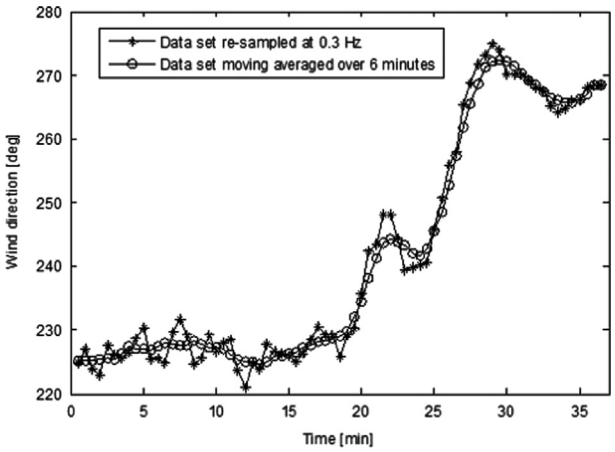

Fig. 5. Example of data set averaged every 30 s.

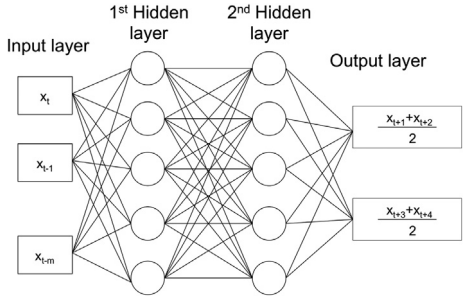

Fig. 6. Feed forward multilayer ANN.

2.3. SVR forecast

SVR model is implemented using the software LIBSVM (Chang and Lin, 2011), together with its Matlab interface. This software has already been used successfully in time series forecasting (Chen et al., 2004). The kernel function is the radial basis function kernel shown in the following equation:

$$k(\overline{x},\overline{y}) = \exp\left(\frac{\|\overline{x}-\overline{y}\|}{2\sigma^2}\right) \tag{6}$$

Similar to the ANN case, a parameter study for different lengths of the input vector and lengths of moving average will be presented in Section 3. It will be shown that there is an optimum for each of these parameters that allows the maximum accuracy of the forecast.

2.4. Performance indices

In order to assess the performance of the forecast we use two different evaluation indices, the mean absolute error and the mean effectiveness index.

The mean absolute error (MAE) is defined as in the following equation:

$$MAE = \langle \hat{x}_t - x_t \rangle \tag{7}$$

where $\hat{x}_t$ is the 30-second-averaged wind direction forecast for the time t, while x_t is the 30-second-averaged wind direction measured at time t, the bar represents the average over the last 100 min of the data set and the $\langle \cdot \rangle$ operator represents the average of 100 ensembles. It should be reminded that, when using ANNs, each forecast is the average over an ensemble of ten ANNs, therefore MAE includes the results of 1000 ANNs. In Section 3, MAE is presented with the 95%-confidence-level error bars computed as two standard deviations of the absolute error over 100 ensembles.

The effectiveness index (EI) is defined to evaluate the performance of the forecast in relation to the specific application that it is intended for. As discussed in Section 1.1, we want to forecast wind shifts larger than 3° for two minutes ahead. In particular, we want to know if the average wind direction for one minute ahead will change by more than 3° and in which direction; and also if and in which direction it will change in the second minute with respect to the first one in order to understand if it is a temporary or a permanent wind shift. Fig. 7 shows a schematic diagram of the effectiveness index. Firstly we measure the average wind direction over the past minute that we use as a baseline. Then we forecast the average wind direction from now to one minute ahead, and from one to two minutes ahead. We compare the average of the first minute ahead with the baseline and we determine if the wind shift is smaller than 3° or, otherwise, the sign of the wind shift. Similarly we do for the average of the second minute ahead with respect to the one of the first minute ahead. Therefore the shift can be negligible, positive or negative for the first minute ahead, and negligible, positive or negative for the second minute ahead. If the forecast is correct for the first minute then the effectiveness index at one minute is one. If the forecast is correct for both the first and the second input, the effectiveness index at two minutes is one. In this case we will be able to recognise those wind shifts where tacking is worthwhile and we will distinguish between temporary and permanent shifts.

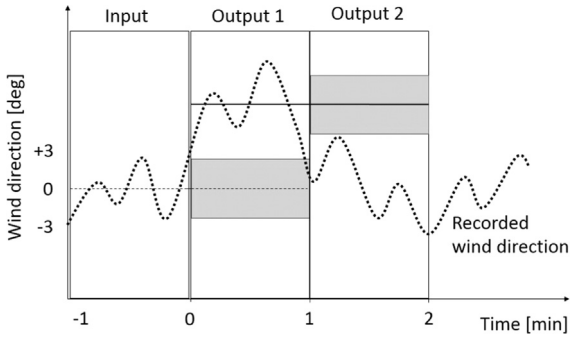

Fig. 7. Effectiveness index.

The mean effective index (MEI) is defined for the ANN as in Eq. (8a), where the bar represents the average over the test set and the $\langle \cdot \rangle$ operator represents the average of 100 ensembles. With the same notation, the MEI for the SVR forecast is defined in the following equation:

$$MEI = \langle \overline{EI} \rangle \text{ for ANN} \tag{8a}$$

$$MEI = \langle EI \rangle \text{ for SVR} \tag{8b}$$

2.5. Test matrix

In order to optimise the network, the MAE and the MEI are computed for different lengths of the moving average, lengths of the input vector and number of neurons in the hidden layers. In particular, moving averages from one to ten minutes, input vector lengths from 4 to 30 values (i.e. from 2 to 15 min) and from 5 to 100 neurons per hidden layer are tested.

The longer the moving average, the smoother the data and the easier to model the trend. However, increasing the length of the moving average leads to a loss of information in the lower frequencies that may result in the inability to forecast low frequency fluctuations. Therefore the optimum moving average length depends on the lowest frequency which needs to be forecast.

The longer the input vector the more information is fed to the model. However feeding unnecessary information makes the convergence of the synaptic weights more difficult to reach and, given a constant length of the available data set, the longer the input vector the fewer the vectors which can be built for training the network. Therefore, while a too-short vector length lacks of the necessary information for the forecast, a too-long vector length results in an undertrained network. It should also be noted that the computational time increases significantly with the length of the input vector.

When using ANN, increasing the number of neurons allows the network to model more complicated nonlinear trends. On the other hand, more neurons lead to more connections and thus to more synaptic weights which have to converge during the training (i.e. more degrees of freedom). Ultimately, for a constant length of available data set, too many neurons lead to nonconverged synaptic weights.

3. Results

In this section, first we present a parameter study of 5 different lengths of the moving average used to pre-process the input data, 14 lengths of the input vector and, for the ANN forecast, also 6 different sizes of layers. Then, using the optimum parameters selected from the parameter study, we compare the performance of the ANN and SVR forecasts.

3.1. ANN optimisation

Figs. 8(a)–10(b) present MAEs and MEIs of one and two minutes ahead for different lengths of the moving average, different lengths of the input vector, and different number of

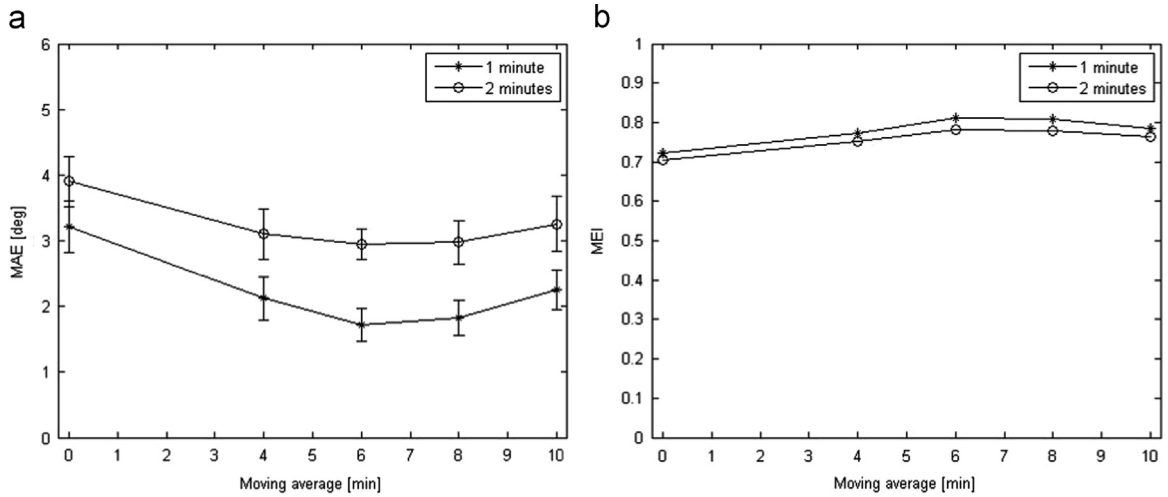

Fig. 8. ANN performance indices versus length of the moving average: (a) Mean absolute error; (b) mean effectiveness index.

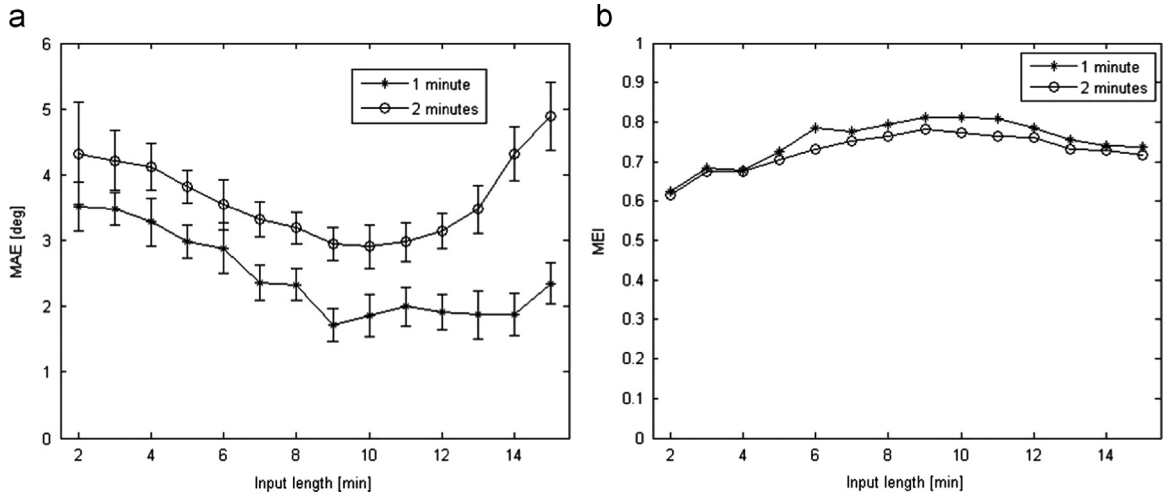

Fig. 9. ANN performance indices versus input length: (a) Mean absolute error; (b) mean effectiveness index.

neurons in the hidden layers. Due to the multidimensionality of the test matrix, the trend of the two performance indices versus each parameter is shown only for the best combination of the other two parameters.

Fig. 8(a) and (b) shows the MAE and the MEI as functions of the length of the moving averages, where each hidden layer has 20 neurons and the size of the input vector is 18 (9 min). Fig. 8(a) shows that for a moving average over 6 min, the MAE of one minute ahead is $1.7^\circ \pm 0.3^\circ$, while the MAE of the two-minute-ahead forecast is $3.0^\circ \pm 0.2^\circ$. As expected, the MAE increases both for smaller and larger lengths of the moving average. Importantly, the MAE of both one and two minutes ahead is minimum for a length of the moving average of six minutes. Fig. 8(b) shows that the two-minute-ahead MEI is 0.78 when a six minute moving average is used, i.e. 78% of the time the forecast is able to predict the correct tactical decision.

Fig. 9(a) and (b) shows the MAE and the MEI, respectively, as a function of the length of the input vector, where a moving average of six minutes are used to smooth the data set and each hidden layer has 20 neurons. Fig. 9(a) shows that the MAE for one and two minutes ahead is a minimum for an input vector size of 18 (9 min) and 20 (10 min), respectively. The MAE does not show a unique optimum input vector size. However, Fig. 9(b) shows that the two-minute-ahead MEI is a maximum for an input vector size of 18 (9 min). Therefore, for this specific application, the best forecast is performed using the past 9 min.

Fig. 10(a) and (b) shows the MAE and the MEI, respectively, as a function of the number of neurons per each hidden layer, where a moving average of six minutes and an input vector size of 18 (9 min) are used. In this case, both indices show a clear optimum when 20 neurons per hidden layer are used.

3.2. Support vector regression

Fig. 11(a) and (b) shows the MAE and MEI, respectively, as a function of the length of the moving average used to pre-process the data, using an input of 8 min. In this case the error bars are not shown, as using different subsets of the data available for training leads to differences in performance that are significantly lower than the ones found in the ANN case (of the order of 10% of the ANN error bars). Overall, the SVR shows a significantly lower MAE, both for one and two-seconds ahead predictions. The best performance is achieved for a moving average over 4 min, allowing a MAE of 0.8° and 1.2° for one and two minutes ahead, respectively, and a MEI of 0.79.

Fig. 12(a) and (b) shows the MAE and MEI, respectively, as a function of input vector for the SVR model, where a moving average of 4 min is used to smooth the data. The lowest MAE is achieved for an input length of 8 min, and is of 0.8° and 1.2° for one and two minutes ahead, respectively. However, the maximum

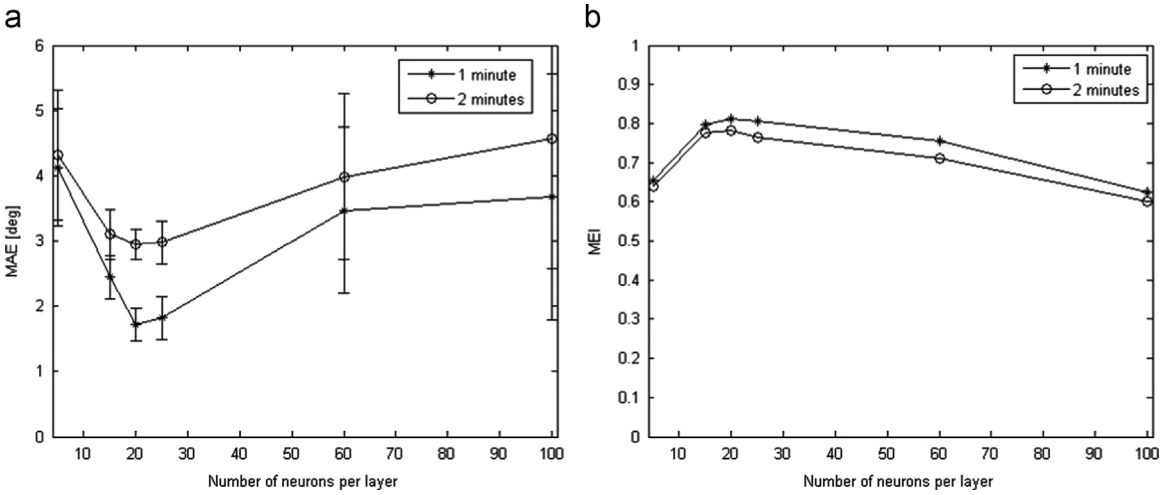

Fig. 10. ANN performance indices versus number of neurons per layer: (a) Mean absolute error; (b) mean effectiveness index.

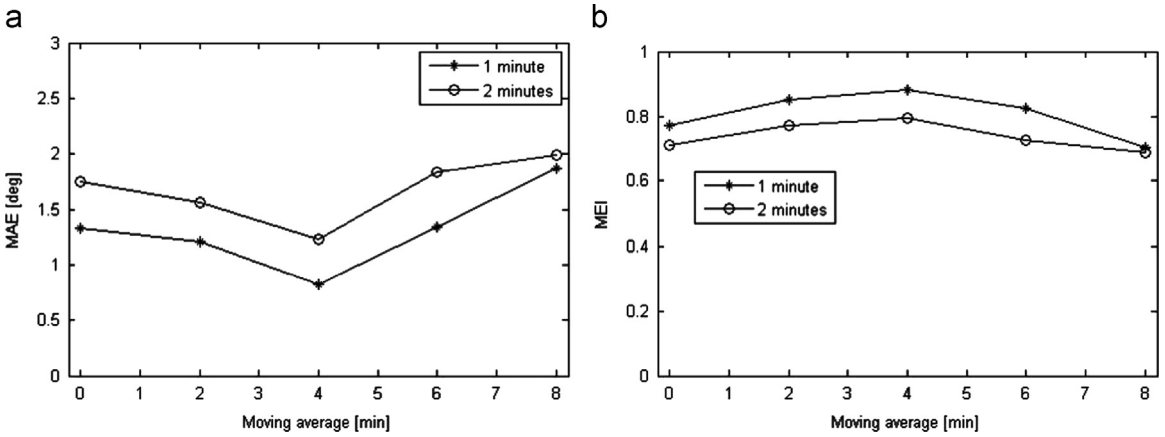

Fig. 11. SVR performance indices versus length of the moving average: (a) Mean absolute error; (b) mean effectiveness index.

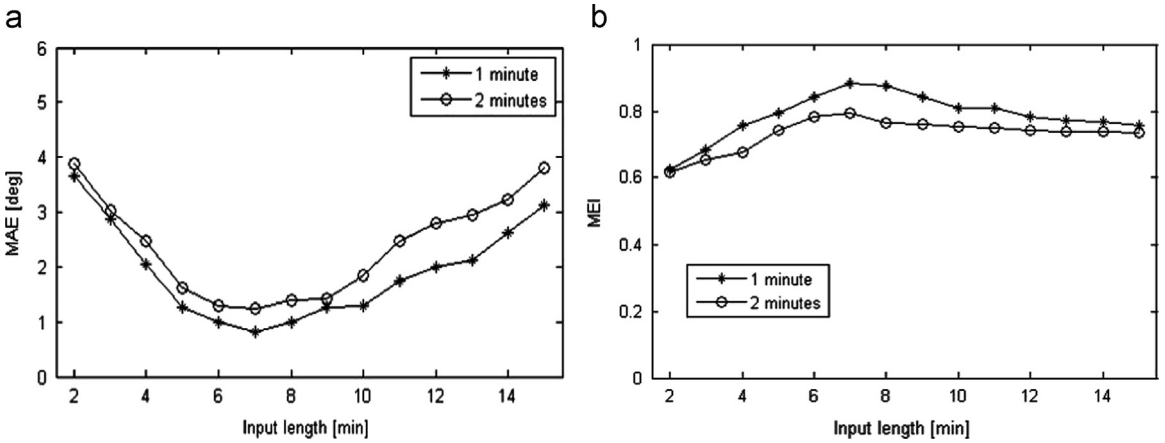

Fig. 12. SVR performance indices versus input length: (a) Mean absolute error; (b) mean effectiveness index.

MEI achievable is of 0.79, differing only by 0.01 from the one achievable with the optimal ANN ensemble forecast.

3.3. *Performance of the optimised ANN and SVR*

The results of the parameter study, which was based on a test set of 100 min, showed that the optimum ANN configuration allows a MAE of 1.7° and 3.0° and a MEI of 0.81 and 0.78 for one and two minutes ahead, respectively, while the optimum SVR allows a MAE of 0.8° and 1.2° and a MEI of 0.88 and 0.79 for one and two minutes ahead, respectively. By performing a statistical test using different subsets of the available data, it is possible to say that the ANN model performs better than the SVR model only with a p-value of 0.09.

In Figs. 13 and 14 we test the optimum configuration for ANN and SVR on the last race of the 34th America's Cup.

The input data set for this race is the one shown in Fig. 5. For clarity, we show only every second forecast, thus one forecast per minute. For each minute, the solid dots show the mean recorded wind direction of that minute, while the stars and the circles show

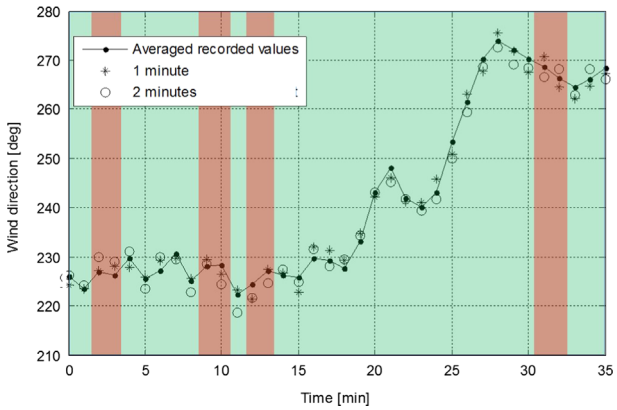

Fig. 13. Example of forecast for the optimised ANN. Red stripes highlight wrong tactical decision, green stripes correct ones. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

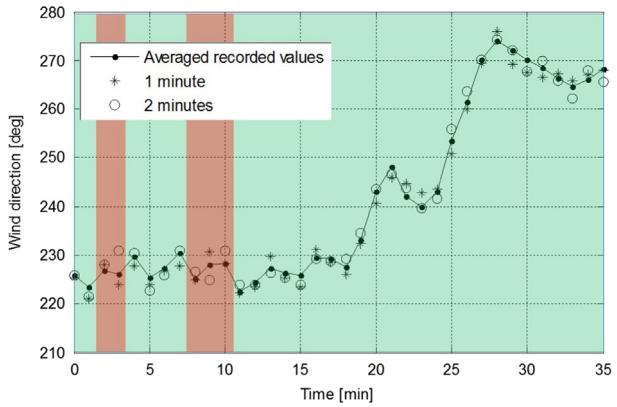

Fig. 14. Example of forecast for the optimised SVR. Red stripes highlight wrong tactical decision, green stripes correct ones. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

the forecast mean wind direction of that minute computed one and two minutes before, respectively. The plots are coloured with vertical bars showing the value of the effectiveness index. Each minute is coloured green (light grey when printed in black and white) if the effectiveness index is one, and is coloured red (dark grey) if zero. Therefore, for each green minute the forecast would have led to the optimum tactical decision, while for each red minute it could have led to a mistake. In particular, every red minute underlines the cases in which the combination of one-minute-ahead and two-minute-ahead forecasts was not accurate enough to predict the correct tactical decision.

Fig. 13 shows that the ANN forecast leads to the correct tactical choice in all cases but four, corresponding to a MEI of 0.86. Fig. 14 shows the SVR forecast on the same set. This forecast leads to three mistaken decisions, corresponding to a MEI, on this particular race, of 0.91.

The choice of obtaining the ANN forecast as an ensemble average of the outputs of ten networks constituted a compromise between training time and computational resources. With adequate hardware and lighter software, the variance of the error could be reduced further. As an example, we tested an ensemble average of 1000 networks (instead of 10) on the test set shown in Fig. 13 and the MEI reached 0.97, corresponding to just one potential mistaken decision.

In conclusion, SVR allows a better forecast in terms of accuracy and computation time, but increasing the computational power it is possible to obtain a better forecast from the combination of ANN models.

4. Conclusions

In this study we present two methods for short term wind direction forecasting based on ANN and SVR. Both methods use the knowledge acquired from previous recordings of wind direction to forecast the near future values. The reliability of the forecast is evaluated by computing the mean absolute error and the mean effectiveness index of the forecast, the latter being an index of the percentage of times in which the forecast is able to predict the correct tactical decision during a sailing yacht race.

The optimum ANN configuration allows a mean absolute error of 1.7° and 3.0°, and a mean effectiveness index of 0.81 and 0.78 for one and two minutes ahead, respectively. The optimum configuration based on the mean absolute error and the one based on the mean effectiveness index are almost identical but for a marginal difference in the optimum length of the input vector. Therefore these results can be generalised to a certain extent to other applications. In particular it is expected that the optimum length of the moving average may decrease if wind shifts averaged over less than one minutes are desired; while longer input vectors and more neurons may be used if a longer data set is available to train the networks.

The optimum SVR forecast allows a mean absolute error of 0.8° and 1.2°. SVR outperforms ANN, both in terms of mean error and computational time. However, the performance in terms of MEI is similar to the ANN model, increasing only from 0.78 for the ANN to 0.79 for the SVR for the two-minutes ahead forecast.

In order to decrease the dependency of the forecast from the training of the ANN, the forecast is made of the ensemble average of ten ANNs subjected to different trainings. The mean effectiveness index was found to significantly increase with the size of the ensemble. For instance, a test performed on the wind recorded during the last race of the 34th America's Cup shows that, increasing the ensemble size from 10 to 1000, the mean effectiveness index increases from 0.79 to 0.97, corresponding to a reduction of the number of potential mistaken decisions from three to one during the entire race. Therefore, with adequate computational resources, the use of large ensembles for ANN forecasts can lead to better performance.

Appendix A. Supplementary data

Supplementary data associated with this article can be found in the online version at http://dx.doi.org/10.1016/j.oceaneng.2014.12.026.

References

America's Cup Event Authority, March 2013. AC34 Race Data Export Format Specification, 1.03.

Barbounis, T., Theocharis, J., Alexiadis, M., Dokopoulos, P., 2006. Long-term wind speed and power forecasting using local recurrent neural network models. IEEE Trans. Energy Convers. 21 (1), 273–284.

Bitner-Gregersen, E., Bhattacharya, S., Chatjigeorgiou, I., Eames, I., Ellermann, K., Ewans, K., Hermanski, G., Johnson, M., Ma, N., Maisondieu, C., Nilva, A., Rychlik, I., Waseda, T., 2014. Recent developments of ocean environmental description with focus on uncertainties. Ocean Eng. 86, 26–46.

Chang, C.-C., Lin, C.-J., 2011. LIBSVM: a library for support vector machines. ACM Trans. Intell. Syst. Technol. 2 27:1–27:27, software available at, URL ⟨http://www.csie.ntu.edu.tw/cjlin/libsvm⟩visited on 01/09/2014.

Chaouachi, A., Nagasaka, K., 2012. A novel ensemble neural network based short-term wind power generation in a microgrid. ISESCO J. Sci. Technol. 8 (14), 2–8.

Chen, B.-J., Chang, M.-W., Lin, C.-J., 2004. Load forecasting using support vector machines: a study on EUNITE Competition 2001. IEEE Trans. Power Syst. 19 (4), 1821–1830.

Costa, A., Crespo, A., Navarro, J., Lizcano, G., Madsen, H., Feitosa, E., 2008. A review on the young history of the wind power short-term prediction. Renew. Sustain. Energy Rev. 12 (6), 1725–1744.

Fierro, R., Lewis, F., 1998. Control of a nonholonomic mobile robot using neural networks. IEEE Trans. Neural Netw. 9 (4), 589–600.

Haykin, S., 1994. Neural Networks: A Comprehensive Foundation. Prentice Hall PTR Upper Saddle River, NJ, USA.
Li, G., Shi, J., 2010. On comparing three artificial neural networks for wind speed forecasting. Appl. Energy 87 (7), 2313–2320.
Mabel, M.C., Fernandez, E., 2008. Analysis of wind power generation and prediction using ann: a case study. Renew. Energy 33 (5), 986–992.
Morgan, N., Bourlard, H.A., 1995. Neural networks for statistical recognition of continuous speech. Proc. IEEE 83 (5), 742–770.
Muller, K.-R., Smola, A., Rätsch, G., Schölkopf, B., Kohlmorgen, J., Vapnik, V., 1997. Predicting time series with support vector machines. Artif. Neural Netw. – ICANN'97 1327, 999–1004.
Park, S., Lee, J., Kim, S., 2004. Content-based image classification using a neural network. Pattern Recognit. Lett. 25 (3), 287–300.
Philpott, A., Mason, A., 26–27 January 2001. Optimising yacht routes under uncertainty. In: 15th Chesapeake Sailing Yacht Symposium. Society of Naval Architects and Marine Engineers, Annapolis, MD, pp. 89–98.
Schölkopf, S.P., Vapnik, V., Smola, A., 1997. Improving the accuracy and speed of support vector machines. Adv. Neural Inf. Process. Syst. 9, 375–381.
Vapnik, V., 2000. The Nature of Statistical Learning Theory. Springer-Verlag, New York, USA.

Ocean Engineering 90 (2014) 149–154

Contents lists available at ScienceDirect

Ocean Engineering

journal homepage: www.elsevier.com/locate/oceaneng

On risk attitude and optimal yacht racing tactics

F. Tagliaferri [a,*], A.B. Philpott [b], I.M. Viola [a], R.G.J. Flay [c]

[a] Institute for Energy Systems, School of Engineering, The University of Edinburgh, United Kingdom
[b] Yacht Research Unit, Department of Engineering Science, The University of Auckland, New Zealand
[c] Yacht Research Unit, Department of Mechanical Engineering, The University of Auckland, New Zealand

ARTICLE INFO

Article history:
Received 14 November 2013
Accepted 30 July 2014
Available online 10 September 2014

Keywords:
Yacht race
Tactics
Risk aversion

ABSTRACT

When the future wind direction is uncertain, the tactical decisions of a yacht skipper involve a stochastic routing problem. The objective of this problem is to maximise the probability of reaching the next mark ahead of all the other competitors. This paper describes some numerical experiments that explore the effect of the skipper's risk attitude on their policy when match racing another boat. The tidal current at any location is assumed to be negligible, while the wind direction is modelled by a Markov chain. Boat performance in different wind conditions is defined by the output of a velocity prediction program, and we assume a known speed loss for tacking and gybing. We compare strategies that minimise the average time to sail the leg with those that seek to maximise the probability of winning, and show that by adopting different attitudes to risk when leading or trailing the competitor, a skipper can improve their chances of winning.

1. Introduction

In this paper we model and analyse the problem faced by a skipper who wants to sail an upwind leg of a yacht race, rounding the mark before his opponent. This problem falls into the category of stochastic shortest-path problems, where the cost function to be minimised is the time needed to reach the mark, and it depends on stochastic quantities such as wind direction. Many problems fall into this category and involve routing for emergency response, both civil (Yamada, 1996) and military (Resch et al., 2003), and applications in logistics (Fleischmann et al., 2004) and transport (Shuxia, 2012). The aim is to find a path between two vertices of a graph such that the sum of its constituent edges, often representing a cost, is minimised. When cost depends on random quantities this becomes a stochastic problem, and the standard objective is to minimise expected costs (where costs include time) (Bertsekas and Tsitsiklis, 1991). For yacht races, models which minimise the expected time to finish, or to reach the next mark, have been studied in a number of papers (Philpott and Mason, 2001; Philpott, 2005). This might be appropriate in fleet races where corrected time over a number of races forms a basis for scoring points. Even so, such scoring systems assign rankings in each race and it is well known that rank-based scoring leads to different incentives than those from performance on average (Anderson, 2012).

As observed in Philpott (2005) rank-based scoring takes its most extreme form in match racing, where the objective is to maximise the probability of arriving before the competing yacht. Indeed the time difference between the two boats is not of interest, as opposed to its sign. In this context, the attitude towards risk of the skipper assumes a greater importance. The aim of this work is to show that by changing the skipper's attitude to risk, it is possible to define a strategy that performs better in match races than strategies aimed at minimising the expected time to finish.

Of course, in most forms of match racing, the interaction between the boats is important. A leading yacht will attempt to cover a trailing yacht, not only for tactical reasons, but also to spill turbulent air on the trailing yacht's sails to reduce their drive. Forcing another boat to tack to avoid a collision is also a tactical ploy to increase a yacht's advantage. In this paper we choose to ignore these effects, as well as assuming identical yachts and crew expertise. This is done for modelling convenience as well as simplicity. By focusing solely only on risk attitude we can see to what extent this is important, other effects being equal.

The paper is laid out as follows. In the next section we describe the model of the yacht and basic sailing strategy for the upwind leg of a match race. We then review dynamic programming as an approach to finding the strategy that minimises the expected time to reach the next mark. The following section shows how this is implemented in a routing model that accounts for different risk attitudes of the skipper. We then present the results of some simulations of the strategies that emerge from the routing model.

* Corresponding author.
E-mail addresses: f.tagliaferri@ed.ac.uk (F. Tagliaferri), a.philpott@auckland.ac.nz (A.B. Philpott), i.m.viola@ed.ac.uk (I.M. Viola), r.flay@auckland.ac.nz (R.G.J. Flay).

http://dx.doi.org/10.1016/j.oceaneng.2014.07.020
0029-8018/

1.1. Sailing strategy

The speed of a sailing yacht depends on the wind speed and on the angle between boat heading and wind direction. It is usually expressed as a polar diagram like the one shown in Fig. 1. The numbers around the semicircle represent different true wind angles, while the radial ones represent the boat speed. The red line corresponds to the plot of boat speed for a particular true wind speed. While no direct course is possible straight into the wind, it is possible to sail upwind with an angle between wind direction and sailed course which is usually between 30° and 50°. Sailing closer to the wind direction (lower angle) makes the course shorter, but when sailing at higher angles a boat is faster. Velocity made good (VMG) is the component of yacht velocity in the wind direction. With a constant wind direction from the top mark, an optimal policy maximises VMG. This is typically attained at a true wind angle of around 40–45° (as in this example). In a polar diagram like the one in Fig. 1, it is possible to find the maximum VMG for a given wind speed by finding the intersection between the polar corresponding to the wind speed and the line perpendicular to the upwind direction. For this reason the common route towards an upwind mark, or in general towards the direction from which the wind blows, is a zigzag route. Such a route requires changes of direction which are called *tacks*. When manoeuvring for a tack, a boat points for a few seconds directly into the wind, therefore causing a temporary decrease in boat speed. If the wind is constant during the race and all over the racing area, trying to do the minimum number of tacks is the best choice. Fig. 2(a) shows two possible routes. In a constant wind, the route on the left is faster because it involves just one tack. Fig. 2(b) shows a situation in which the wind shifts towards the left over the duration of the leg. The best policy in this case is to go to the left of the course (referred to as being on *starboard tack*), and then tack and point towards the mark, while a myopic policy that begins the race going to the right (referred to as being on *port tack*) turns out to be suboptimal.

In real races the evolution of the wind can be much more complicated than these examples, with temporary shifts or gusts that a sailor seeks to take advantage of. Moreover wind has a random component. While racing, it is difficult to know how the wind is behaving at another location, or to foresee how it will behave once that point is reached. In the presence of randomness the optimal course in Fig. 2(b) might turn out to be worse than a myopic policy that tacks on every wind shift. For this reason sailors tend to try and stay in the centre of the course to enable shifts in wind direction to be exploited by tacking, while avoiding the risk of overlaying the mark.

In the presence of a competitor, a policy that avoids the course boundaries while staying close to the competitor reduces the risk of being beaten, at least when the competitor is the trailing boat. On the other hand, when the competitor is leading, it can make sense for a skipper to take a risk and explore the corners of the course hoping for a favourable wind shift. This is the phenomenon that we seek to model in this paper.

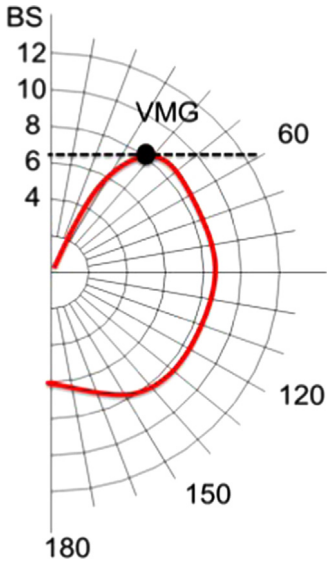

Fig. 1. Example of a polar diagram (velocities in m/s and angles in degrees). (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

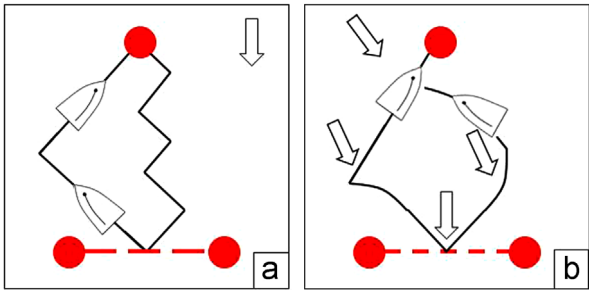

Fig. 2. Example of upwind routes. (a) Constant wind and (b) left wind shift.

1.2. Dynamic programming

Finding an optimal set of tacks when the wind varies randomly requires a *stochastic dynamic* optimisation model. In contrast to the deterministic case, a solution does not consist of a single optimal path for a specific wind realisation, but a *policy* that is optimal over a range of wind realisations. Policies can be computed a priori and respect the principle of optimality: an optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision (Bellman, 1957). A policy that respects this principle can be found with *dynamic programming* (Bertsekas, 1995). Dynamic programming has been successfully applied in sailing in both ocean races and short course racing (see Philpott and Mason, 2001; Philpott, 2005). In this work we adapt the short-course model described in Philpott and Mason (2001) and Philpott (2005) with the aim of incorporating the skipper's attitude towards risk in their actions.

The risk that a skipper is willing to take is usually influenced by his position with respect to the opponent. A common behavioural pattern is to be conservative, or *risk averse*, when in a leading position, while being *risk seeking* when losing. Here we interpret risk aversion as being pessimistic about wind shifts, believing that any shifts we observe will not be to our advantage. In contrast, a risk-seeking skipper will be optimistic about wind shifts and act as if these are more likely to be to his advantage. Such attitudes can be modelled by altering the transition probabilities of the process that defines wind shifts.

To understand the effect of risk-averse or risk-seeking skippers, we develop a race modelling program (RMP) for simulating races between two boats. The first RMP was developed in 1987 for the America's Cup syndicate Stars and Stripes and is described in Letcher et al. (1987). Since then, RMPs have been used mainly in America's Cup applications to compare different designs (see e.g. Philpott et al., 2004). In our case, since we are interested in comparing tactical choices, we model two identical boats (i.e. they have the same polar diagram).

2. Method

2.1. Dynamic programming

We consider an upwind leg of 6000 m (corresponding to 3.24 nautical miles, which approximates the length of the 2013 America's Cup course), and 4000 m wide. In the coordinate system used the starting line is located on the x-axis, and centred around the origin, while the upwind mark is located on the y-axis. The racing area is discretised into a rectangular grid with $N=20$ increments Δx across the course and $M=400$ increments Δy in the direction of the course, as shown in Fig. 3. The $N-1$ lines defining the grid that are perpendicular to the y-axis will be referred to in the following as "cross sections". The dynamic program is at stage i when the yacht crosses the ith cross section.

The state variables are the yacht's position x_i, the wind direction w_i observed at stage i, and the current tack z (where $z=0$ denotes starboard tack and $z=1$ denotes port tack). The wind direction w_i is random and satisfies the *Markov* property, namely that the probability distribution for the variable w_i, conditioned on all the previous values, is equal to the distribution for the variable w_i conditioned just on the last event:

$$\begin{aligned}&\mathbb{P}(w_i = v|w_{i-1} = v_{i-1}, w_{i-2} = v_{i-2}, \ldots, w_0 = v_0)\\ &\quad = \mathbb{P}(w_i = v|w_{i-1} = v_{i-1})\end{aligned} \tag{1}$$

for every $i>0$ and for every w_i in the state space.

The actions at each stage are whether to tack the boat (i.e. change z to $1-z$) or continue on the same tack. As mentioned in the Introduction, a tacking manouvre implies a time loss that will be denoted as τ. Given a yacht's polar and its location, we can compute $t(i,x,x',w,z)$, defined to be the time to sail from location $(x,i\Delta y)$ to $(x',(i+1)\Delta y)$ if it is on tack z and the observed wind direction is w.

We define the value function $T_i(x_i,w_i,z)$ to be the minimum expected time to sail from location x_i on cross section i to the top mark given wind observation w_i, and current tack z. Clearly $T_M(x,w_i,z)=0$ when location x is at the top mark, and we choose $T_M(x,w_i,z)=\infty$ otherwise.

We compute $T_0(x_0,w_0,z)$ for (x_0,w_0,z) corresponding to the boat's position and tack on the start line, using a dynamic programming recursion. First define at stage i the function

$$F(i,x,x',w,z) = t(i,x,x',w,z) + \mathbb{E}_{w'}[T_{i+1}(x',w',z)|w], \tag{2}$$

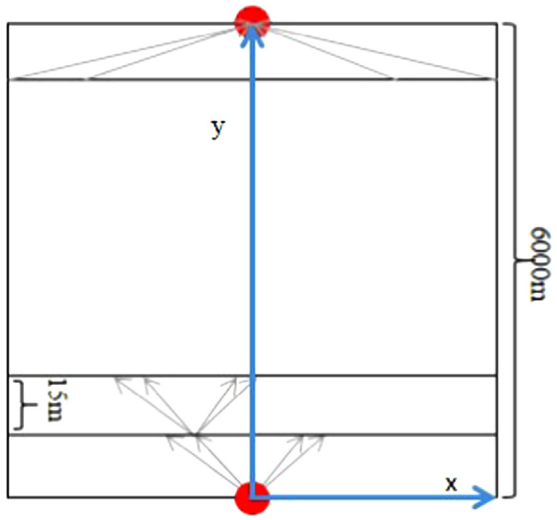

Fig. 3. Schematic representation of the course.

where w' is the wind direction that is observed at stage $i+1$. Now we can define the recursion as follows:

$$T_i(x_i,w_i,z) = \min\begin{cases}\min\limits_{x_{i+1}\in\mathbb{X}} F(i,x_i,x_{i+1},w_i,z)\\ \tau + \min\limits_{x_{i+1}\in\mathbb{X}} F(i,x_i,x_{i+1},w_i,1-z)\end{cases} \tag{3}$$

where $\mathbb{X}$ is the set of x coordinates of positions $(x_{i+1},(i+1)\Delta y)$ that can be reached at stage $i+1$ from position x_i at stage i. More details on the recursive procedure defined by Eqs. (2) and (3) can be found in Philpott and Mason (2001).

2.2. Wind modelling

We assume the wind speed to be constant during the race, focusing on the changes in wind direction. As discussed in the previous section the dynamic programming algorithm we use assumes that the wind direction satisfies the Markov property. Although more refined wind models are being developed (see for instance the recent reviews by Costa et al., 2008 and Bitner-Gregersen et al., 2014), Markov models are computationally very efficient and can still capture most of the statistical properties that are relevant in certain applications (Shamshad et al., 2005; Sahin and Sen, 2001).

For tactical purposes we are interested in changes in wind direction that significantly affect the racing time. We therefore define a finite number of wind direction states: namely -45°, -40°, …, 0°, $+5^\circ$,…, $+45^\circ$, where 0° represents the wind direction at which the upwind mark is set, and the other states represent shifts of $\pm 5^\circ$ from that direction.

For a system with a finite number of states the stochastic process is uniquely defined with an initial distribution for w_0 and a transition matrix $\boldsymbol{P}$. The matrix elements P_{jk} represent the probability that the system at time step i is in state k conditioned on the fact that it was in state j at the previous time step $i-1$:

$$P_{jk} = \mathbb{P}(w_i = k|w_{i-1} = j)$$

In order to obtain a realistic transition matrix we considered a time series of wind measurements from a weather station installed on the Newcastle University research vessel, and then built the matrix $\boldsymbol{P}$ using a maximum likelihood estimator. As we use for the model a grid with 15 m resolution in the upwind direction and the decisions are taken every time the yacht reaches a cross section, the wind is modelled using a time step of 3 s, which is the time spent on average to move between two consecutive cross sections. The recorded wind direction signal was sampled every three seconds, and the corresponding wind directions were placed in K bins of amplitude 5°. The number of jumps from bin j to bin k divided by the total number of jumps out of bin j defines the value $P_{jk}, j,k=1,2,\ldots,K$, in the transition matrix.

Given a transition matrix $\boldsymbol{P}$, Eq. (2) becomes

$$F(i,x,x',w_j,z) = t(i,x,x',w_j,z) + \sum_{k=1}^{k=K} P_{jk} T_{i+1}(x',w_k,z). \tag{4}$$

2.3. Risk modelling

We now turn our attention to the risk attitude of the yacht skipper. There is an enormous literature on modelling risk (for a recent introduction see Anderson, 2013). To model risk aversion, we adopt an approach based on the theory of *coherent* risk measures (Artzner et al., 1999). As shown in Artzner et al. (1999) coherent risk measures can be expressed as the worst-case expectation over a convex set of probability distributions to give a risk-adjusted expectation. Given the current wind direction state,

the probability distribution that we work with is the corresponding row of the transition matrix. To model risk aversion we choose the worst possible transition probabilities from a convex set $\mathcal{D}$ of transition matrices. In other words, (4) becomes

$$F(i,x,x',w_j,z) = t(i,x,x',w_j,z) + \max_{P\in\mathcal{D}} \sum_{k=1}^{k=K} P_{jk} T_{i+1}(x',w_k,z). \quad (5)$$

An interpretation of (5) is illuminating. A boat skipper who is winning will be risk averse. She will try to behave safely, trying to stay ahead and to minimise her losses in bad wind outcomes. Using (5) in a recursion is pessimistic about the next wind shift and assigns a higher probability to the worst outcomes (i.e. heading shifts). Being pessimistic about random outcomes reduces risk, at some loss in expected performance.

Risk seeking behaviour has been less well studied, although it is often given as an explanation for participation in lotteries and negative expectation gambles, where optimistic participants place greater weight on winning probabilities than their real values. In our context we model risk seeking by choosing the best possible transition probabilities from a convex set $\mathcal{D}$ of transition matrices. In other words, (4) becomes

$$F(i,x,x',w_j,z) = t(i,x,x',w_j,z) + \min_{P\in\mathcal{D}} \sum_{k=1}^{k=K} P_{jk} T_{i+1}(x',w_k,z). \quad (6)$$

This has the following interpretation. A boat skipper who is losing will seek risk. If she adopts a minimum expected finish time strategy against another skipper who minimises his expected time to finish, then she will tend to make the same decisions (unless the boats see very different winds) and lose the race almost certainly. She will instead seek different wind conditions from the competitor. Using (6) in a recursion will be optimistic about the possible advantageous wind shifts and assign a higher probability to these outcomes (i.e. lifting shifts). Being optimistic about random outcomes increases risk, as well as incurring some loss in expected performance.

We implement (5) and (6) in the recursion by adding a transformation in the solver that post multiplies the transition matrix by another matrix which redistributes the probabilities. The resulting matrix has to be normalised in order to represent again a probability distribution.

3. Results

Fig. 4 shows a graphical representation of the transition matrix for the Markov model obtained with the maximum likelihood estimator as described in the previous section. With a notation that will be used throughout this paper, we use a grey scale to represent values in the interval [0, 1] where white represents 0 and black represents 1. It can be noticed that the diagonal is dominant, meaning that, in general, if the wind is in state *i*, the most probable state for the next step is to remain in state *i*. Moreover, when the wind has deviated from the mean, the event of a shift back towards the mean value is more likely than one in the same direction.

The wind for the simulations was generated as described in the previous section. The Markov chain defines a discrete wind direction. This can be made continuous by superimposing a mean-reversion noise process (see Philpott et al., 2004). However we did not do this as we found that the behaviour of the simulated wind signal, achieved with no additional noise component, was similar to the empirical one, as can be seen in Fig. 5, with close values of mean and variance on different sub-intervals. A wind history of 400 values was generated for each of the 4000 simulated races.

Fig. 6 shows a histogram of the time needed by a yacht following the policy generated to minimise the expected time of arrival, according to the wind distribution previously modelled. The distribution is asymmetric, and this is due to the fact that even with a very favourable evolution of the wind there is a minimum time needed to complete the course. On the other hand, even with a policy which is effective in the majority of the cases, it is possible to be very unlucky and need a much higher time.

This policy was generated using a risk-neutral transition matrix for wind direction as pictured in Fig. 4. When the skipper is risk seeking or risk averse we replace this with a modified transition matrix. A sailor who is losing will seek risk. This corresponds to increasing her confidence of a lifting wind shift while discounting the likelihood of a heading wind shift. The transition matrices we use to represent a risk-seeking skipper are shown in Fig. 7(a) and (b). As shown in the figures, advantageous shifts (cells below the diagonal when the skipper is to the left of the opposition, and cells above when on the right) happen with higher probability than in the risk-neutral case. The remaining probabilities in each row are reduced to add to one.

The transition matrices for a risk-averse skipper are constructed similarly. Here bad wind shifts (above the diagonal when the skipper is to the left of the opposition, and below the diagonal

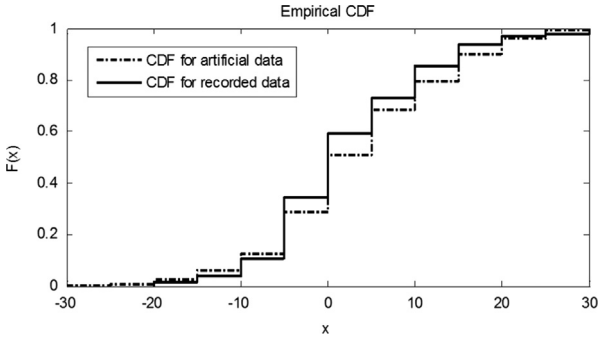

Fig. 5. Sixty-minute example of artificially generated wind and sixty-minute example of recorded wind.

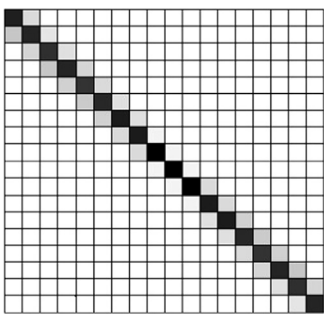

Fig. 4. Representation of the transition matrix obtained for the wind model.

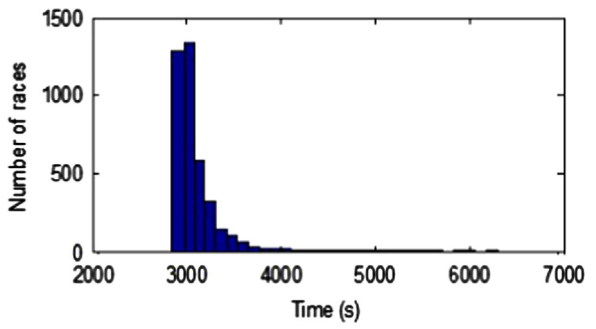

Fig. 6. Distribution of arrival time of boat following the optimum policy.

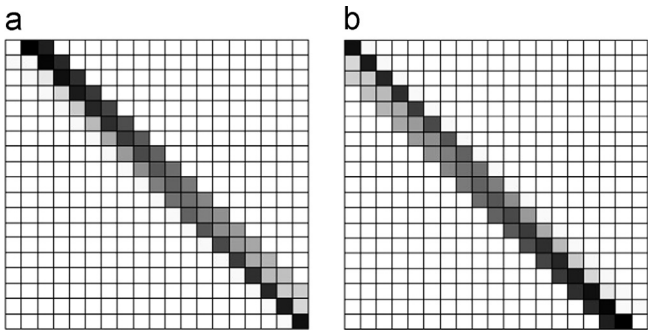

Fig. 7. Modified transition matrices for a risk-seeking skipper. Advantageous wind shifts occur with higher probability than disadvantageous ones. (a) Yacht on the left-hand side of competitor and (b) yacht on the right-hand side of competitor.

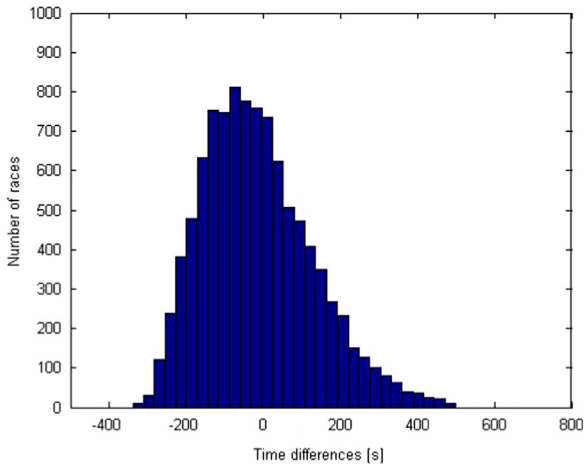

Fig. 8. Histogram of arrival time of B minus arrival time of A.

when on the right) happen with higher probability than in the risk-neutral case. In our experiments we have obtained transition matrices for a risk-averse policy by simply swapping the matrices in Fig. 7(a) and (b).

Simulations were carried out in order to verify the differences between a risk-neutral policy that minimises expected arrival time at the top mark, and a policy generated assuming either risk seeking or risk averse behaviour. Results showed that policies that minimise expected arrival time won more races than either being consistently risk seeking or risk averse.

However, combining the strategies together (to allow both risk-seeking and risk-averse behaviour at different times) can lead to a significant improvement in the chances of winning. We simulated races between two boats that are denoted as boat A and boat B. Both boats start the race at the same time, on two different (random) points along the starting line. Boat A experiences the simulated wind and always follows the risk-neutral policy (to minimise expected arrival time). Boat B experiences the same wind as A if their distance apart is less than $d_{min} = 10$ m, an independent wind if their distance is greater than $d_{max} = 100$ m, and a linear combination of A's wind and an independent sample if their distance is between d_{min} and d_{max}. At every step of the simulation, if B is more than 15 s behind A, she uses the risk-seeking policy depending on the side of the course; if B is more than 20 s ahead of A, she uses the risk-averse policy, while she uses the optimum risk-neutral policy otherwise. Results of those simulated races are shown in Fig. 8.

The x-axis shows the arrival time of boat B minus the arrival time of boat A at the top mark. The average time difference is positive (actually 16 s in this plot). This means that B arrives 16 s later on average than A, as one would expect, since A is using the optimum policy to minimise the average time. However about 63% of the race outcomes are to the left of zero, meaning that B wins 63% of the time (always by a small margin). Of course sometimes B is hopelessly outclassed, losing by 400 s (just around 0.01% of the times, and those are extremely unfavourable events) but this is because B takes high risks when behind. If we consider $p=0.5$ win probability as a null hypothesis, then the probability of winning more than 63% of 5000 races by chance is the probability that a binomial random variable with mean $5000p$ and variance $5000p(1-p)$ exceeds 3150, which is negligible.

The standard error of the value 0.63 can be estimated using the central limit theorem to be approximately 0.0035. So we can be 97.5% confident that the hybrid policy will win at least 62.3% of the races (i.e. 2 standard errors less than 0.63).

In order to quantify the tactical improvement on the policy we compare the results obtained by boat A and boat B with a third boat C that has perfect knowledge of the future behaviour of the wind. In this case we simulated 1000 races. Obviously the boat with perfect knowledge of the wind scenario always wins and the increases in arrival time of A and B are always positive. The sample average difference in time of arrival is 133 s for boat A while for boat B the sample average difference is 149 s. The difference is not significant because of high variance and low sample size. However this experiment confirms a theoretical result: the expected time difference for boat A relative to C is never more than the expected time difference for boat B relative to C (see Appendix for proof).

4. Conclusions

In this paper we have presented a method for approximating a solution of a stochastic shortest path problem with applications to yacht racing. We showed that with an adequate subdivision of the problem it is possible to find a solution that minimises the expected time needed to reach an upwind mark during a race.

Moreover, we introduce for the first time a model of the risk attitude of the sailor. We showed that if a skipper of a trailing boat has a risk-seeking attitude it enhances the chance to win the race. An important result of the simulations run to simulate races was that aiming at minimising the expected time to finish is not always the best approach: being on average slower might allow a bigger probability of winning against an opponent following a fixed policy.

The results presented in this paper underline that when trying to optimise a policy in order to win a competition, looking at average values is rarely the best approach, and accounting for differing risk attitudes might give policies that perform significantly better. Further work is being carried out in order to validate the model with data registered during America's Cup races, and we are developing methodologies for learning risk parameters that yield maximum win probabilities.

Acknowledgements

This research has been performed within the SAILING FLUIDS project (PIRSES-GA-2012-318924), wich is funded by the European Commission under the 7th Framework Programme through the Marie Curie Actions, People, International Research Staff Exchange Scheme. The authors would like to thank Newcastle University for providing wind data.

Appendix

Proposition 1. *Minimising the expected arrival time over all strategies will give a policy that is slower than a perfect skipper by the least amount on average.*

Proof. Suppose a perfect skipper sails races in wind that she predicts perfectly. Each race is a random sample of wind and so her time to finish is an independent identically distributed random variable T.

Suppose she now sails a strategy s that is not clairvoyant in each of these same wind conditions. The time to finish under this strategy is an independent identically distributed random variable $S(s)$.

Now the delay in finishing under strategy s versus the perfect strategy is also an independent identically distributed random variable $D(s) = S(s) - T$. The expected delay from sailing s is then

$$\mathbb{E}[D(s)] = \mathbb{E}[S(s)] - \mathbb{E}[T].$$

To minimise this we should minimise $\mathbb{E}[S(s)]$ as $\mathbb{E}[T]$ is a constant. So the strategy that minimises expected delay after a clairvoyant skipper is the one that minimises expected arrival time. □

References

Anderson, E., 2013. Business Risk Management: Models and Analysis. John Wiley & Sons Inc., United States, ISBN: 978-1-118-34946-5.

Anderson, E.J., 2012. Ranking games and gambling: when to quit when you're ahead. Oper. Res. 60 (5), 1229–1244.

Artzner, P., Delbaen, F., Eber, J.-M., Heath, D., 1999. Coherent measures of risk. Math. Financ. 9 (3), 203–228.

Bellman, R.E., 1957. Dynamic Programming, 1st edition Princeton University Press, Princeton, NJ, USA.

Bertsekas, D.P., 1995. Dynamic Programming and Optimal Control. Athena Scientific, Belmont, MA, United States.

Bertsekas, D.P., Tsitsiklis, J.N., 1991. An analysis of stochastic shortest path problems. Math. Oper. Res. 16 (3), 580–595.

Bitner-Gregersen, Elzbieta M., Bhattacharya, Subrata K., Chatjigeorgiou, Ioannis K., Eames, Ian, Ellermann, Katrin, Ewans, Kevin, Hermanski, Greg, Johnson, Michael C., Ma, Ning, Maisondieu, Christophe, Nilva, Alexander, Rychlik, Igor, Waseda, Takuji, 2014. Recent developments of ocean environmental description with focus on uncertainties. Ocean Engineering 86, 26–46. http://dx.doi.org/10.1016/j.oceaneng.2014.03.002, ISSN 0029-8018.

Costa, A., Crespo, A., Navarro, J., Lizcano, G., Madsen, H., Feitosa, E., 2008. A review on the young history of the wind power short-term prediction. Renew. Sustain. Energy Rev. 12 (6), 1725–1744.

Fleischmann, B., Gnutzmann, S., Sandvoß, E., 2004. Dynamic vehicle routing based on online traffic information. Transp. Sci. 38 (4), 420–433.

Letcher, J.S., Marshall, J.K., Oliver, J.C., Salvesen, N., 1987. Stars and Stripes. Sci. Am. 257, 34–40.

Philpott, A.B., 2005. Stochastic optimization and yacht racing. In: MOS-SIAM Series on Optimization, SIAM, Philadelphia, PA, United States, pp. 315–336.

Philpott, A.B., Henderson, S.G., Teirney, D., 2004. A simulation model for predicting yacht match race outcomes. Oper. Res. 52 (1), 1–16.

Philpott, A.B., Mason, A.J., 2001. Optimising yacht routes under uncertainty. In: The 15th Cheasapeake Sailing Yacht Symposium, Annapolis, USA, pp. 89–98.

Resch, C.L., Piatko, C., Pineda, F.J., Pistole, J., Wang, I.-J., 24 March 2003. Path planning for mine countermeasures. in: AeroSense, Orlando, Florida. International Society for Optics and Photonics, pp. 1279–1286.

Sahin, A.D., Sen, Z., 2001. First-order Markov chain approach to wind speed modelling. J. Wind Eng. Ind. Aerodyn. 89 (34), 263–269.

Shamshad, A., Bawadi, M.A., Hussin, W.M.A.W., Majid, T.A., Sanusi, S.A.M., 2005. First and second order Markov chain models for synthetic generation of wind speed time series. Energy 30 (5), 693–708.

Shuxia, L, 2012. Study on routing optimization problem of the logistics center. In: World Automation Congress (WAC), IEEE, Puerto Vallarta, Mexico, 24–28 June 2012, pp. 1–4.

Yamada, T., 1996. A network flow approach to a city emergency evacuation planning. Int. J. Syst. Sci. 27 (10), 931–936.

FEEDFORWARD NEURAL NETWORKS FOR VERY SHORT TERM WIND SPEED FORECASTING

F Tagliaferri and **I M Viola,** Yacht and Superyacht Research Group, School of Marine Science and Technology, Newcastle University, UK

SUMMARY

Since 2007 wind has become the major source of renewable energy in the UK. Moreover, increasing oil costs are driving researchers in the marine transport field to develop innovating wind ships. In order for wind power to be effectively and efficiently exploited, reliable forecasts on wind speed are needed. These will allow saving curtailments costs, improving safety, reducing damages due to extreme weather conditions, etc. Also, short and very short wind forecasts are critical for energy trading. In this study we present a short-term wind forecast based on artificial neural networks, which are mathematical structures able to model complex non-linear systems. In particular, we used a multilayer perceptron that predicts future wind speed values given the past and current recorded values. Data sampled every ten minutes was used to forecast up to one hour ahead, with an uncertainty ranging from 5%, for ten minutes ahead forecast, to 21%, for one hour ahead forecast.

1. INTRODUCTION

The 2011 annual wind report of the World Wind Energy Association (WWEA) [1] stated that, in 2011, the worldwide wind energy production increased by 40,053 MW, reaching a total capacity of 237,016 MW, which corresponds to the biggest increase in history, but also corresponding to a growth rate of 20,3% which is the lowest in the last decade The European Union has approved in 2008 the climate and energy package (known as 20/20/20 strategy) whose aim, by the year 2020, is to reduce greenhouse gas emissions by 20%, to establish a 20% share for renewable energy, and to improve energy efficiency by 20% [2]. In order to meet this challenging target, deployment and operation of wind energy device need a significant step change. In particular, there is an unmet need for wind forecasting. More specifically, while long term forecasts (up to five days) allow decisions for deployment and maintenance of the structure; short (up to 24 hours) and very short (up to 1-2 hours) are critical for energy trading [3,4].

There are two possible approaches to wind power forecasting: one is to predict wind velocity and then use a power curve to convert it to energy production; another one is to predict directly the wind energy. The choice between these two approaches needs to take into account that also the relationship between wind speed and wind energy production can have a stochastic nature [5] or being highly nonlinear [6].

In this work we present a model for the prediction of wind velocity. Models for predicting wind speed can be numerical or statistical. Numerical model are based on mathematical fluid mechanics models and have dominated the literature until the last decade (for instance, [7,8]). Statistical methods are based on past observation of the wind behaviour at one or more locations and have been found to perform better than numerical models on very-short-term forecasts [9,10]. In this paper we present a statistical model based on Artificial Neural Networks (ANN) using past recorded values of wind speed to predict the future values.

ANN have been successfully used to attack a wide range of problems such as, for instance, speech recognition [11], image classification [12], function approximation [13] and financial forecasting [14]. The present study uses ANN to predict the future wind velocity based on several successive sets of velocity measurements taken at a single location.

Differently from earlier studies using ANNs for wind forecasting [15], where temperatures, humidity and pressures were also input to the model, our model uses only the recent past velocities allowing a fast training of the network.

2. METHOD

ANN are inspired by the functioning of the biological neural networks in the brains of humans and animals, and their peculiarity is the possibility of emulating the human process of learning from experience. The constitutive unit of a neural network is a neuron, which is a singular processing unit that takes several inputs x_i originating from other neurons, and produces an output that is then transmitted to other neurons. The mathematical representation of the structure of a neuron is shown in Figure 1. A neuron itself can be broken down into the following components:

- A set of connecting links, called synapses, where the $i-th$ synapses is characterized by a weight w_i (synaptic weights);
- An adder within the neuron that performs a sum of the inputs weighted by the corresponding synapses;
- An activation function φ, which transforms the sum computed by the adder into the neuron output. If the activation function is linear, a neuron results in a linear combination of the input values, while non-linear activation functions (generally sigmoid functions) allow for the modelling of non-linear problems.

Therefore, a neuron can mathematically be described by Equation (1):

$$y = \varphi(\zeta);\ \ \zeta = \sum_{i=1}^{u} w_i x_i \qquad (1)$$

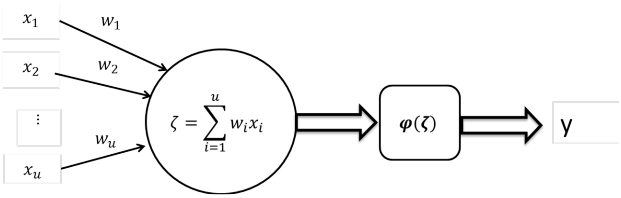

Figure 1. Structure of a generic neuron.

Neurons are assembled together into an integrated structure that depends on the kind of problem that the network has to solve. A structure that has been successfully used in function approximation is the so-called feed-forward multi-layer perceptron, characterized by the organization in subsequent layers of the neurons, as can be seen in the schematic example in Figure 2.

The learning process involves the continuous modification of the synaptic weights and it is based on the principle of iterative error-correction. The synaptic weights of the various neurons are initialized to random values, then a training set of input and output data is presented to the network.

For each input vector the initially generated output vector is compared with the known true output vector. The synaptic weights of the output layer are then modified by adding a factor that is proportional to the current assessed error and to a learning rate, and those corrections are extended to all of the weights in the network through a back-propagation process. This operation is iterated until successive changes in the synaptic weights are smaller than a given value, or when the errors begin to increase. For further details on training algorithms and validation processes see [16].

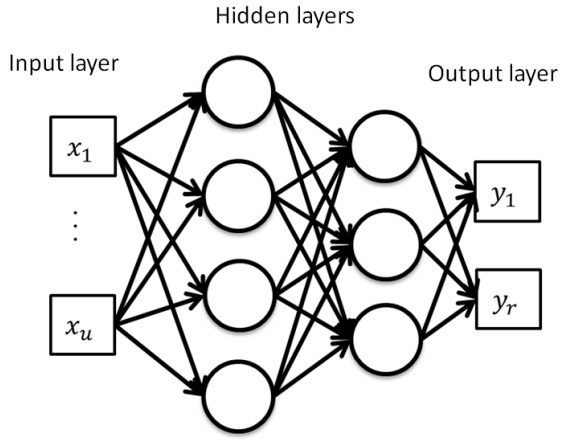

Figure 2: Schematic diagram of a multi-layer feed-forward perceptron.

In this work we use a multilayer perceptron to perform a forecast for wind speed based on past values. A time series approach is used: we assume that there exists a function f such that:

$$s(t_{k+1}) = f\big(s(t_k), s(t_{k-1}), \dots, s(t_{k-m})\big) \qquad (2)$$

where $s(t)$ represents the wind speed at time t. In the present study the time values for the sampling $t_k$are taken at a distance of 10 minutes.

Therefore, we are looking for a way to express the next future value for the wind speed as a function of a vector of past values. The ANN is trained in order to model the function f. An input vector, consisting of consecutive measured values for the wind speed, is used as input, while future values are used as training outputs for the network.

3. RESULTS

A multi-layer perceptron was used to perform a forecast on ten-minutes wind speed measurements. The data set, provided by the National Climate Database and available online [17], was made of 4000 consecutive measured values; 80% of the values were used for the training, while the remaining 20% were used for testing the performance of the trained ANN.

Different networks were trained in order to identify the best structure to perform the forecast.

Single layer perceptrons can be used only to model linearly separable problems [16], and tests confirmed this limitation as it was observed that the performance of such a network was highly dependent on the initialization of the synaptic weights. Conversely, two hidden layers allow modelling non-linear functions, such as f in Eq. (2). Increasing both the number of neurons per layer and the size of the input vector - increasing m in Eq. (2) - led the average error to decrease until optimum values are reached. Then a further increase in neurons led the performance to decrease again. Both an excessive number of neurons and a too large size of the input vector led the performance to decrease because the number of parameters to be optimised increases and the training becomes inefficient. For instance, for our data set, the best performance was achieved with two hidden layers with 18 and 15 neurons, respectively, and with an input vector made of eight consecutive wind speed measurements - $m = 7$ in Eq. (2). The single output vector predicted the ninth consecutive value.

Having two different non-linear activation functions for the two layers increases the learnability when dealing with highly non-linear models [16]. For instance, we used a log-sigmoid and an hyperbolic tangent activation function for the first and second layer, respectively.

The training is performed with the Levenberg-Marquardt back-propagation algorithm, which is known to be efficient for networks with less than 100 neurons [18].

Figure 3 shows the comparison between wind speed values registered by the station and wind speed forecast

by the ANN. The signal corresponding to the wind speed is highly oscillating and no qualitative general trend can be extrapolated. However, the ANN is able to perform a forecast with an uncertainty of 5% at 95% confidence level (meaning that, in 95% of the cases, the error is less than 5% of the average the wind speed).

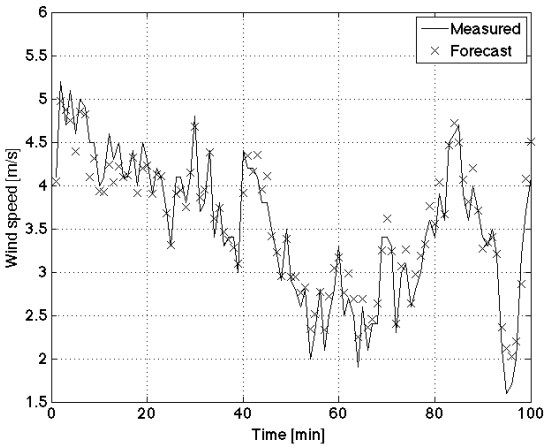

Figure 3: Measured and forecast wind speed.

The same method with a different training can be used in order to predict the successive-step-ahead wind speed value. Therefore, being $s(t_k)$ the most recent measured wind speed known by the ANN, the input vectors are still of the form $[s(t_k), s(t_{k-1}), \dots, s(t_{k-m})]$, while the output is $s(t_{k+j})$, with j varying from 0 to 6. We use again $m = 7$. The forecast uncertainty increases with j. In particular, Table 1 shows the uncertainty for the different minutes ahead forecasts. Ten minutes ahead is computed with $j = 1$, and one hour ahead is computed with $j = 6$.

Table 1: Uncertainty for different minutes ahead forecasts.

Minutes ahead [min]	10	20	30	40	50	60
U [%]	5	7	11	15	18	21

4. CONCLUSIONS

Reliable short and very short wind energy production forecasts are necessary for energy trading. In order to obtain a good forecast it is possible to transform a forecast for wind speed into a forecast for wind energy production for a wind farm. In this paper, we presented a model for short-term wind forecast based on artificial neural networks. The model uses ten-minutes wind measurement and, taking as input eight consecutive wind values, predicts the wind speed for the next values up to one hour ahead. The uncertainty increases almost linearly with the time distance between the last input values and the desired output. The model allows forecasting up to one hour ahead with an uncertainty of 21%.

5. REFERENCES

1. GSANGER, S., PITTELOUD J. D., 'WWEA Report 2011', 2011.

2. EUROPE Press Release: "Climate change: Commission welcomes final adoption and energy package", available at the web page http://europa.eu/rapid/pressReleasesAction.do?reference=IP/08/1998.

3. PARKES, J., WASEY, J., TINDAL, A., MUNOZ, L., "Wind Energy Trading Benefits Through Short Term Forecasting" *European Wind Energy Conference technical Papers,* 2006.

4. POTTER, C. W., "Very short-term Wind Forecasting for Tasmanian Power Generation" *IEEE Transactions on Power Systems,* 2006.

5. JEON, J., TAYLOR, J. W., 'Using Conditional Kernel Density Estimation for Wind Power Density Forecasting', *Journal of the American Statistical Association,* 2012.

6. SANCHEZ, I., "Short-Term Prediction of Wind Energy Production", *International Journal of Forecasting, 22, 43-56, 2006.*

7. COTTON W. R., PIELKE R. A., WALKO R. L., LISTON, G. E., TREMBACK, C. J., JIANG, H., MCANELLY, R. L., HARRINGTON, J. Y., NICHOLLS, M. E., CARRIO, G. G., MCFADDEN, J. P., "RAMS 2001: Current status and future directions" *Meteorology and Atmospheric Physics, 82 (1-4), 5-29*, 2003.

8. JACOB D., PODZUN, R., "Sensitivity studies with the regional Climate Model REMO", *Meteorology and Atmospheric Physics 63(1-2), 119-129,* 1997.

9. LI, S., "Using neural networks to estimate wind turbine power generation," *IEEE Trans. Energy Convers.*, *16(3), 276–282,* 2001.

10. ALEXIADIS M. C., DOKOUPULOS P. S., SAHSAMANOLOGOU H. C., "Wind Speed and Power Forecasting Based on Spatial Correlation Models," *IEEE Trans. Energy Convers.*, *14(3), 836–842,* 1999.

11. MORGAN N., BOURLARD H. A. (1995), "Neural Networks for Statistical Recognition of Continuous Speech", *Proceedings of the IEEE, 83 (5), 742-770.*

12. LAWRENCE S., GILES C.L., TSOI A. C., BACK A.D., "Face Recognition: A convolutional Neural-Network Approach",

IEEE Transactions on Neural Networks, 8(1), 98-113 1997.

13. POGGIO T, GIROSI F., "Networks for Approximation and Learning", *Proceedings of the IEEE, 78(9), 1481-1497*, 1990.

14. KAASTRA I., BOYD M., "Designing a Neural Network for Forecasting Financial and Economic Time Series", *Neurocomputing, 10(3), 215-236*, 1996.

15. MORE A., DEO M. C., "Forecasting Wind with Neural Networks", *Marine Structures, 16(1), 36-49,* 2003.

16. HAYKYN S., "Neural Networks, A Comprehensive Foundation", *Prentice Hall PTR Upper Saddle River NJ USA,* 1999.

17. NIWA, National Climate Database, http://cliflo.niwa.co.nz/, visited on 10th May 2012

18. HAGAN, M., MENHAJ, M. B., "Training Feedforward Networks with the Marquard Algorithm", *IEEE Transactions on Neural Networks, 5(6), 989-993,* 1994.

6. AUTHORS BIOGRAPHY

Francesca Tagliaferri is a PhD student at Newcastle University and member of the Yacht and Superyacht Research Group. She holds a Masters degree with Honours in mathematics and her PhD project aims at developing a navigation software for yacht races under uncertain weather conditions.

Dr. Ignazio Maria Viola is Lecturer in Naval Architecture at the Newcastle University and group leader of the Yacht and Superyacht Research Group. His specialist skills and experience are in numerical and experimental incompressible viscous fluid dynamics. In particular, his main research focus is the hydro and aero dynamics of yachts and superyachts, while his research portfolio also includes the fluid dynamics of renewable energy devices, trains, cars, long-span bridges and tall buildings. In 2008 he successfully completed a CFD simulation with a record breaking grid of more than one billion cells and in 2010 he performed the world's first direct pressure measurements on a full-scale spinnaker. Ignazio has more than 40 peer reviewed scientific publications on the fluid dynamics of yachts and super yachts.