UPPSALA
UNIVERSITET

UPTEC F15053
Examensarbete 30 hp
September 2015

Modeling, control and state-estimation for an autonomous sailboat

Jon Melin

UPPSALA
UNIVERSITET

Teknisk- naturvetenskaplig fakultet
UTH-enheten

Besöksadress:
Ångströmlaboratoriet
Lägerhyddsvägen 1
Hus 4, Plan 0

Postadress:
Box 536
751 21 Uppsala

Telefon:
018 – 471 30 03

Telefax:
018 – 471 30 00

Hemsida:
http://www.teknat.uu.se/student

Abstract

Modeling, control and state-estimation for an autonomous sailboat

Jon Melin

During long time missions with autonomous sailboats electrical power is a limited resource. When making evaluations of power saving strategies it is important to have a good platform for simulations and experiments. In this thesis a model of a four meters long autonomous sailboat is presented and parameterized, several control strategies are developed and evaluated and state estimation algorithms for the wind and the sailboat are developed, all with the purpose of future studies of power management.

The model that is used captures the main characteristics of a sailboat quite well, it is possible to use the model for controller development and state estimation. The model and the controllers are evaluated using simulations and experiments. The control strategy and state estimation algorithms shows promising results, but more experiments are needed to verify that simulations and the experiments are consistent in all conditions.

Handledare: Matias Waller
Ämnesgranskare: Thomas Schön
Examinator: Tomas Nyberg
ISSN: 1401-5757, UPTEC F15053

Acknowledgements

First of all, I would like to express my gratitude to my supervisor at Åland University of Applied Science Matias Waller, for your support during this project. I'm especially grateful for all the discussions we have had and I have always felt welcome to come to you with my thoughts. Kjell Dahl also deserves a great thank, it has been pure enjoyment to work with you.

I would also like to thank Anna Friebe and Ronny Eriksson at Åland University of Applied Science for making it possible for me to take part in the project Åland Sailing Robots and for inspiring discussions about autonomous sailing.

Thanks to my subject reviewer at Uppsala University, Thomas Schön for introducing me to the beauty of control theory and the Åland Sailing Robots project.

Furthermore, all the members at the Åland Curling Club deserves a thank for all the entertaining evenings at the rink.

Finally I would like to thank friends and family for their support and understanding.

Contents

Acknowledgements iii

List of Figures vii

List of Tables viii

1 Introduction 1
- 1.1 Åland Sailing Robots . . . 2
- 1.2 Microtransat challenge . . . 2
- 1.3 Motivation . . . 3
- 1.4 Outline of thesis . . . 4
- 1.5 Associated publication . . . 4
- 1.6 System description . . . 5
  - 1.6.1 Hardware . . . 5
  - 1.6.2 Software . . . 7

2 Modeling 9
- 2.1 Reference systems, states and variables . . . 10
- 2.2 Model equations . . . 10
- 2.3 True and apparent wind . . . 11
- 2.4 Acting forces . . . 13
  - 2.4.1 Sail force . . . 13
  - 2.4.2 Rudder force . . . 14
  - 2.4.3 Friction . . . 14
- 2.5 Model simulations . . . 14
- 2.6 Experiments . . . 15
  - 2.6.1 GPS coordinates and great circles . . . 16
- 2.7 Comparison of simulation and experiment . . . 17

3 Control 21
- 3.1 Control strategy . . . 22

3.1.1 Rudder control 23
3.2 Rudder control evaluation 24

4 State estimation **29**
4.1 Sailboat filter 30
4.1.1 Discretization of model equations 30
4.1.2 Measurement equations 31
4.1.3 Extended Kalman filter equations 31
4.2 Wind filter 32
4.3 Filter implementation and results 33

5 Concluding remarks **37**
5.1 Conclusions and discussion 37
5.2 Further work 39

Bibliography **40**

List of Figures

1.1 The sailboat and parts of the team 1
1.2 Experimental system overview 5
1.3 Electronic system 6
1.4 Remote radio control 7
1.5 Raspberry Pi 2 Model B v1.1 7
1.6 XBee module 7
1.7 Wind sensor 7

2.1 Degrees of freedom of ships 10
2.2 Explanation of the states in the model 11
2.3 Definition of wind 12
2.4 Trajectory. Test case Circle 18
2.5 Speed profile. Test case Circle 18
2.6 Trajectory. Test case Straight line 19
2.7 Speed profile. Test case Straight line 19
2.8 Trajectory. Test case Tack 19
2.9 Speed profile. Test case Tack 19

3.1 Comparison of rudder angle as a function control error 25
3.2 Rudder comparison simulation 26
3.3 Comparison of rudder angle as a function control error 27
3.4 Trajectory from experiment with controller 28

4.1 True wind direction 34
4.2 True wind speed 34
4.3 Trajectory from state estimation, $p_4 = 200\text{kg}\,\text{s}^{-1}$ 35
4.4 Trajectory from state estimation, $p_4 = 500\text{kg}\,\text{s}^{-1}$ 35

List of Tables

2.1 Model parameters 12
2.2 Model parameters and their values 20

3.1 Parameters used in the controller 23
3.2 Performance evaluation of rudder control 25

Chapter 1

Introduction

Figure 1.1: Åland Sailing Robots sailboat and parts of the team during a test sailing in May 2015. A laptop for monitoring, data acquisition and calculation of control signals are located in the escorting boat. *(Included in the thesis with permission from Åland University of Applied Science. Foto: Therese Andersson)*

Humans have sailed for thousands of years and simple methods for rudder control have been used for a long time, such as fixation of the tiller or the more advanced wind vane self steering system, both of them are mechanical solutions. The invention of electronic gyrocompasses was an important step towards automatic steering and making automatic steering common during the first half of 20th century. Systems for automatic control of sails have not been studied as long and was first started to be researched around 1990. Most of the available research focuses on controlling the sails relative angle to the wind, other parts of sail control, e.g. reefing and avoidance of luffing needs more attention. The interests for autonomous

sailing has increased lately but there are few commercial products available, one example is the Saildrone[1], which focuses on marine research. Other potential applications for autonomous sailing with small vessels are for example surveillance, minefield mapping, unmanned ferrying and CO_2-neutral transportation of gods. Still many difficulties needs to be resolved, [2, 13, 14].

1.1 Åland Sailing Robots

Åland Sailing Robots (ÅSR)[2] is a project at the Åland University of Applied Sciences concerned with autonomous sailing. The main goal of ÅSR is to take part in the competition Microtransat challenge and be the first team to succeed in crossing the Atlantic ocean. It is also an important part of the project to contribute to the research and knowledge on green technology and autonomous vessels. ÅSR focuses on autonomous sailing with small sailboats that has a length of 1 to 4 meters.

1.2 Microtransat challenge

The Microtransat challenge is a competition for fully autonomous sailboats and aims to stimulate development of autonomous sailing in a friendly way. The goal of the challenge is to cross the Atlantic ocean, either from west to east or from east to west. The maximum waterline length of boats taking part in the competition is 4 meters, but due to legal reasons this length will be decreased to 2.4 meters in 2017 [12]. Several teams from different countries have made attempts to cross the Atlantic ocean but so far none has succeeded. Two attempts has been made in 2015, in march started Team Joker from Ascot, Berkshire in United Kingdom. Their boat Snoopy sailed for two days but was mostly drifting with the tide. Snoopy capsized before making it out of the English channel [10]. In June started United States Naval Academy their sailboat ABoat Time, it sailed for almost 8 days before it stopped functioning [12]. Although it sailed for quite some time it seems to have had problems already from the start since its heading always where about 20° too far to the north.

[1] `http://www.saildrone.com`
[2] `http://www.sailingrobots.ax`

1.3 Motivation

When working with autonomous sailboats that should sail for weeks or month without coming to harbour power management is important. A small sailboat have limited capacity to carry solar panels and accumulators and therefore also a limited capacity to supply the control and measurement system with electricity. A related challenge is power management for the control unit, sensors and actuators. A separate study from ÅSR presents detailed power management solutions for operating different sensors and actuators at different, and varying, sampling rates as well as the microcontroller at different, and varying, clock rates [1]. A part of the solution is to shut down the measurement and control system completely or partially for longer periods. Informed and safe choices to shut down and wake up the electronic system must be based on reliable estimates of the state of the boat, i.e., mainly the position and heading. An important question to answer is if improved estimates of position could be obtained at modest computational expense compared to the use of dead reckoning, i.e., constant speed and heading.

For further investigations on power consumption using the methods in [1] is a functioning and flexible system for experiment useful. Therefore, in this project, the focus is not on a practical implementation of the overall strategy but rather on a flexible solution for evaluating different aspects of the strategy. It lies not in the scope of this project to determine if and how big the power savings could be. From literature a model will be selected/developed and evaluated with simulations and experiments. The model will also be used for state estimation and closed loop simulations.

A model will be selected for the sailboat and evaluated using three criteria:

- How well does the model describe the behaviour of the Mini 12 used for the practical trials?
- Can the model be used to develop a useful control strategy?
- Can the model be used for reliable state estimation?

The reason for selecting these three criteria is connected to the overall strategy for the long-term operation of an autonomous sailing vessel discussed in the above mentioned study [1].

1.4 Outline of thesis

In section 1.6, the sailboat used in this project is described, including the boat itself, the electronics and the software. In chapter 2 a sailboat model is presented, parametrized and evaluated. This chapter also includes a description of how simulations and experiments were performed. In chapter 3 a control strategy is presented and evaluated. Chapter 4 includes the theory necessary for implementing the filters used for state-estimation and some results indicating what the filters might be capable of. Chapter 5 includes conclusions and a discussion about the results presented in the previous chapters together with recommendations for further work.

1.5 Associated publication

In addition to this report the work performed in this thesis did also result in a paper, *Modeling and control for an autonomous sailboat: A case study*, accepted for publication and presentation at the 8th International Robotic Sailing Conference in Mariehamn, Åland Islands, August 2015 [11]. The paper was written in cooperation with Kjell Dahl and Matias Waller. Some formulations in this report is taken directly from the paper.

1.6 System description

The sailboat used for the experiments in this project is shown in figure 1.1, it has an overall length of just over 4 meters. The sailboat is of the model Mini-12 and also qualifies to the 2.4mR-class of the International Sailing Federation. It weighs about 300 kg. The maximum angle the rudder can make with the keel is 30° and the maximum angle the main sail can make is about 35°, this limitation is due to the way the actuator controlling the main sheet is installed. Meaning that the actual maximum sail angle is larger.

Although a primary goal of the project is to have an autonomous and energy-efficient measurement and control system, the experimental setup used in this project is designed for easy supervision and real-time evaluation. To achieve this it is necessary to have a way of transferring information, in real time, from the sensors on board the boat to a laptop in a nearby boat. Therefore a wireless link was installed during this project. The other hardware, presented next, was already installed prior to this project.

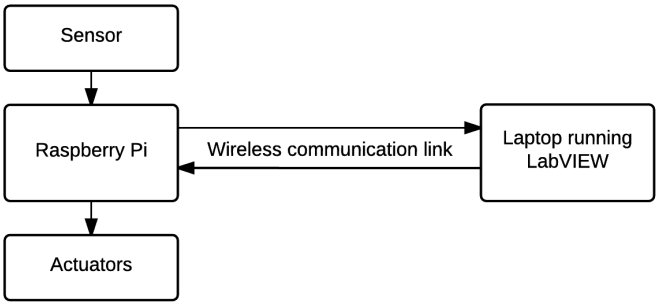

Figure 1.2: Experimental system overview, the on-board system to the left communicating wireless with a laptop.

1.6.1 Hardware

The hardware setup used during this project, figure 1.2, mainly uses the system on board the sailboat for reading sensors and writing signals to the actuators. Calculations of control signals to the actuators as well as presentation of sensor readings is done on the laptop.

A diagram of the electronic system on board the sailboat is provided in figure 1.3. The main unit in the system is a Raspberry Pi (2 model B v1.1), a small single board computer, figure 1.5. The Raspberry Pi collects data from all sensors

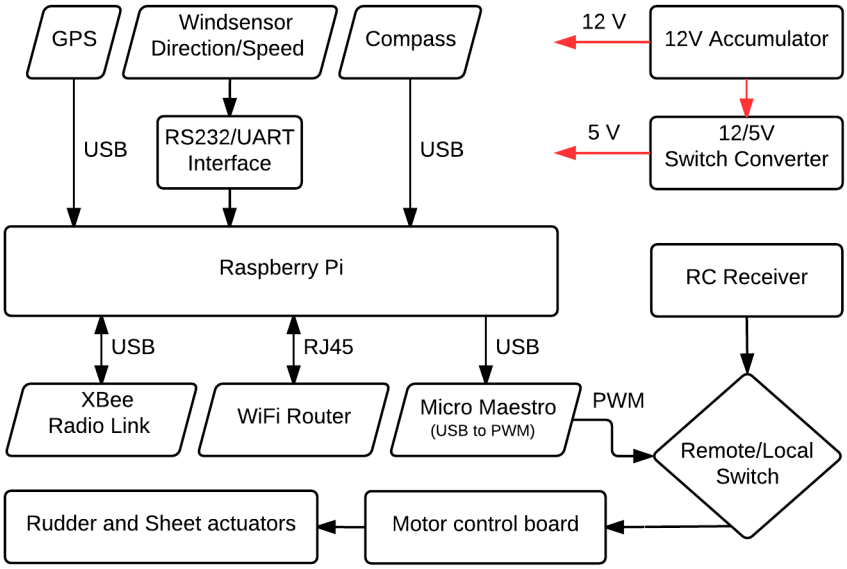

Figure 1.3: Electronic system on board the sailboat: Components and inter-component communication.

and then transmits it to the external laptop via a 2.4 GHz radio link using a XBee module (XBP24Z7SIT-004), figure 1.6. The XBee module is also used for receiving control signals to the actuators. The Raspberry Pi can not send control signals directly to the actuators instead is a USB-PWM board used for this communication (Polulu Maestro Servo Controller). The Raspberry Pi send signals via USB to the board that convert them to PWM[3] signals. The PWM signals controls the actuators via two motor control boards (Polulu Jrk 12v12). Before reaching the motor control boards are the signals passing a relay switch (Pololu RC Switch with Relay). The other input and the control of the relay is connected to a traditional radio control system (6 channel Futaba 6J-2.4GHz), figure 1.4. This is for safety reasons, if for some reason the Xbee link or the Raspberry Pi fails, it is possible to switch the PWM signals to originate from a traditional radio hand controller, and thus bypassing as much of the electronic system as possible. The system also includes a wifi router (Dovado GO 3G) making it possible to log on to the Raspberry Pi over wifi, e.g. to make changes to the program source code. The range of the wifi network is short, only a few meters. Two similar but slightly different linear actuators from Concentric International are used to control the rudder and the main sheet. The actuators are connected via ropes and pulleys to the rudder and sheet. The actuator (LACT8P-12V-20) for the main sheet is shorter and stronger then the actuator for the rudder (LACT12P-12V-5).

[3]PWM - Pulse width modulation

Figure 1.4: Remote radio control used for manual control between experiments and as backup if the system fails.

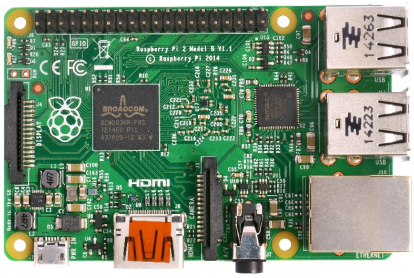

Figure 1.5: The main unit in the electronic system, Raspberry Pi 2 Model B v1.1.

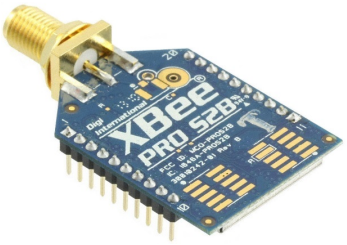

Figure 1.6: XBee module (XBP24Z7SIT-004), a 2.4 GHz radio link module used for communication between the on-board system and the external laptop.

Figure 1.7: Wind direction/speed sensor (LCJ Capteurs CV7) mounted in the top of the sailboats mast.

The sensors available on the sailboat are, a GPS (GlobalSat BU-353), aCompass (HMC6343) and a wind direction/speed sensor (LCJ Capteurs CV7), figure 1.7.

1.6.2 Software

The Raspberry Pi is running Arch Linux, and the source code for the sailboat is written in C++ and is open source.[4]

The laptop has a XBee unit connected to it, and thus receiving all sensor readings sent from the sailboat. On the laptop LabVIEW[5] is installed. In LabVIEW three versions of a sailingrobot application has been developed. First, *Monitor and log*,

[4]Code is available at `https://github.com/pophaax/`

[5]LabVIEW - A software design and development platform with capabilities to connect sensors developed by National Instruments.

which is presenting all sensor readings in a GUI and storing every received message in a text file. The second version *Manual control*, does the same thing and also enables the possibility to send control signals given by the user to the sailboats actuators. This means that both the control signals sent and the control signals used at every time instance will be logged. Due to communication delays, these two signals differ slightly. The third version *Controller* is the most complex version, its main purpose is to implement a controller by letting the laptop calculate the control signals. This version is made in such a way that it is possible for the user to easy and in real time switch between and manual and automatic control of rudder and main sheet separately. It also includes the possibility choose how and if the wind should be filtered. The parts of the application that filters the wind and calculates the control signals is implemented with a MATLAB[6]-script block, enabling the use of the same MATLAB functions written for simulations, thus is the risk of implementation errors of these steps in the applications very small. It is possible for the user to turn of the filter and manually set values for wind directions and speed. Altogether this makes the application a good tool for testing new controllers that has been developed and simulated in MATLAB.

[6]MATLAB - numerical computing environment developed by MathWorks

Chapter 2

Modeling

For the purpose of navigation and control of ships, a rather exhaustive presentation of models can be found in the book by Fossen [3]. The book is mainly about ships and underwater vehicles, but a similar approach to modelling has been extended to sailboats in [15]. The model presented in [15] has also been used as a basis for theoretical studies on simulations of, and control strategies for sailboats [6, 9]. A simplified model has been presented in [8], with a modest number of parameters and thus an attractive alternative for control design and state estimation. Given the success of practical trials for autonomous tracking by a sailboat achieved by the team around Jaulin, it seems that the simple model captures the dynamics of a sailboat essential for controller design. Therefore, the model presented in [8] is also chosen as the basis for the model development of the present project.

It is necessary to make some initial assumptions in order to keep the model simple.

- Motion is confined to a horizontal plane at see level without roll or pitch. This gives a model with 3 degrees of freedom: surge, sway and yaw defined in figure 2.1.
- Influence from waves and currents will be neglected.
- Velocity is assumed to be small, therefore Coriolis and Centripetal forces are neglected. Also the damping coefficients could be assumed to be linear of the same reason.
- The mainsail and the foresail is combined into one effective sail

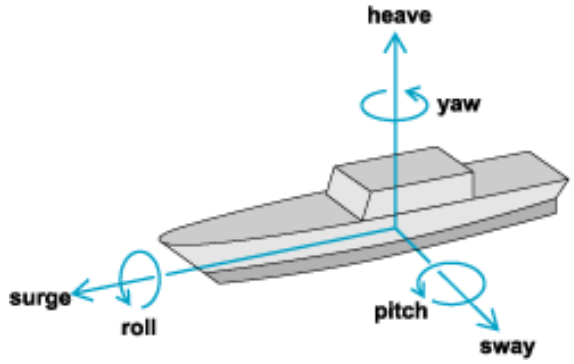

Figure 2.1: Movement of ships is often refereed to as surge, sway, heave, roll, pitch and yaw, e.g. 6 degrees of freedom.

2.1 Reference systems, states and variables

In order to represent all states of the sailboat and measurements, several reference systems are needed. The sailboats position, x, y, heading, θ, and rotational speed, ω, is given in a North-East-Up reference frame (n-frame), i.e. an easterly x-axis and northerly y-axis, figure 2.2. The n-frame is earth fixed and assumed to be inertial for local navigation and its origin in simulations and experiments will be set to the sailboats starting point, e.g. the n-frame is a tangential plane to the surface of the earth. For the experiments in this project is this approximation sufficient, but when sailing over longer distances, such as the Atlantic ocean, the curvature of the earth must be taken into consideration.

It is also necessary to define a sailboat fixed reference frame (b-frame). The origin of the b-frame lies in the boats center of gravity (CoG) which also is assumed to coincide with the boats center of rotation (CoR), figure 2.2. In the b-frame is the sailboats speed, v, distances on the sailboat and the control signals $\boldsymbol{u} = \begin{bmatrix} \delta_r & \delta_s \end{bmatrix}^{\mathrm{T}}$ defined. δ_r is the angle of the rudder and δ_s is the angle of the sail which is proportional to the length of the mainsheet.

2.2 Model equations

The model is based on traditional translational and rotational inertia affected by forces, yielding changes in position, orientation, speed and rotation. In equation (2.2) the model is given by non-linear differential equations in state space with five states, $\boldsymbol{x} = \begin{bmatrix} x & y & \theta & v & \omega \end{bmatrix}^{\mathrm{T}}$, defined in section 2.1. Drift and change in position

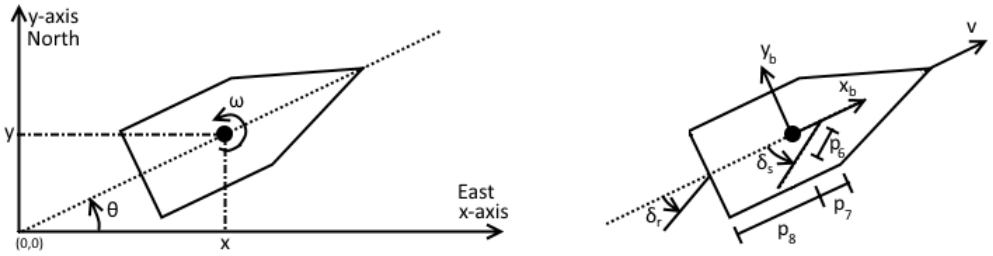

Figure 2.2: Left: Position (x, y) and orientation (θ) of the sailboat is given in a North-East-Up reference frame. Right: Velocity (v) and angle of rudder (δ_r) and the angle of the sail (δ_s) is given in a sailboat fixed reference system together with distances (p_6, p_7 and p_8).

is described in the first two lines of equation (2.2). Drift occurs when wind interacts with all surfaces except the sails, e.g., hull and mast. The acceleration of the sailboat, line 4 in equation (2.2), is affected by three forces: The propulsion from the sails, a breaking force from the rudder and a tangential friction force. The rotational acceleration, line 5 in equation (2.2), follows from differences in moments.

In compact notation the model is given by,

$$\dot{\boldsymbol{x}} = m(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{W}_{p,\text{tw}}) \tag{2.1}$$

corresponding to,

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{v} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} v\cos(\theta) + p_1 a_{\text{tw}} \cos(\psi_{\text{tw}}) \\ v\sin(\theta) + p_1 a_{\text{tw}} \sin(\psi_{\text{tw}}) \\ \omega \\ (g_s \sin(\delta_s) - g_r p_{11} \sin(\delta_r) - p_2 v^2)/p_9 \\ (g_s(p_6 - p_7\cos(\delta_s)) - g_r p_8 \cos(\delta_r) - p_3 \omega v)/p_{10} \end{bmatrix} \tag{2.2}$$

where $\boldsymbol{u}$ is the control signals defined in section 2.1, $\boldsymbol{g} = \begin{bmatrix} g_s & g_r \end{bmatrix}$ is the forces on sail and rudder defined in section 2.4 and $\boldsymbol{W}_{p,\text{tw}}$ is the true wind defined in section 2.3. An explanation of all the parameters p_i can be found in table 2.1. The distances p_6, p_7 and p_8 are defined in figure 2.2 and further explained in table 2.1.

2.3 True and apparent wind

Wind is described both in the n-frame and in the b-frame corresponding to true wind (tw) and apparent wind (aw) respectively. True wind is the velocity and

[7] CoE - center of effort

Table 2.1: Model parameters

p_1 -	drift coefficient	p_7 [m]	distance to mast
p_2 [kgs^{-1}]	tangential friction	p_8 [m]	distance to rudder
p_3 [kgm]	angular friction	p_9 [kg]	mass of boat
p_4 [kgs^{-1}]	sail lift	p_{10} [kgm^2]	moment of inertia
p_5 [kgs^{-1}]	rudder lift	p_{11} -	rudder break coefficient
p_6 [m]	distance to sail CoE[7]		

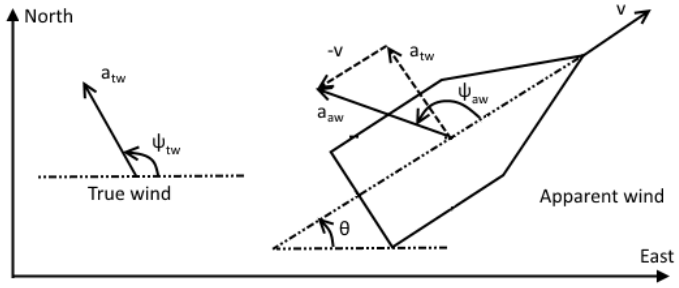

Figure 2.3: Definition of true wind, wind relative an earth fixed point, and apparent wind, wind relative the sailboat.

direction of the air measured from a platform fixed to the ground, e.g. the wind as it is experienced by a person standing still on a pier. Apparent wind is the velocity and direction of the wind measured from a moving object, e.g. the wind as it is experienced by a person standing at the fore of a ship. Figure 2.3 illustrates the angle ψ and speed a of the wind in polar coordinates in both frames. For convenience $\boldsymbol{W}_{p,\mathrm{tw}} = \begin{bmatrix} a_{\mathrm{tw}} & \psi_{\mathrm{tw}} \end{bmatrix}^{\mathrm{T}}$ is introduced. Given the speed and heading of the boat, true wind can be calculated from the apparent wind or vice versa. Apparent wind in Cartesian coordinates relative to the direction of the boat, i.e., the first coordinate corresponding to the heading of the boat can be calculated from true wind by

$$\boldsymbol{W}_{c,\mathrm{aw}} = \begin{bmatrix} a_{\mathrm{tw}} \cos(\psi_{\mathrm{tw}} - \theta) - v \\ a_{\mathrm{tw}} \sin(\psi_{\mathrm{tw}} - \theta) \end{bmatrix} \tag{2.3}$$

The corresponding polar coordinates are thus given by

$$\boldsymbol{W}_{p,\mathrm{aw}} = \begin{bmatrix} a_{\mathrm{aw}} \\ \psi_{\mathrm{aw}} \end{bmatrix} = \begin{bmatrix} |\boldsymbol{W}_{c,\mathrm{aw}}| \\ \mathrm{atan2}(\boldsymbol{W}_{c,\mathrm{aw}}) \end{bmatrix} \tag{2.4}$$

where atan2 is the arctangens function with two arguments returning an angle in the correct quadrant.

2.4 Acting forces

It is possible to apply basic aerohydrodynamic theory if sail and rudder are regarded as thin foils. This means that the forces acting on the sail and rudder due to the medium flowing around them could be simplified into a lift and a drag force acting in a center of effort, equation (2.5). The drag force is acting co-linear to the apparent wind and lift force is acting perpendicular.

$$\begin{aligned} L &= \frac{1}{2}\rho A v_{\mathrm{a}}^2 C_L \\ D &= \frac{1}{2}\rho A v_{\mathrm{a}}^2 C_D \end{aligned} \tag{2.5}$$

where ρ is the density of flowing medium, A is the plane area of the foil, v_{a} is the mediums apparent velocity, C_L and C_D are lift and drag coefficients respectively. These coefficients depends on the mediums angle of attack, α and the foils shape.

In [15] it can be seen that the drag and lift coefficients could be approximated by,

$$\begin{aligned} C_{\mathrm{L}} &= k_1 \sin(2\alpha) \\ C_{\mathrm{D}} &= k_1(1 - \cos(2\alpha)) \end{aligned} \tag{2.6}$$

where k_1 is a constant. This means that the lift and drag forces can be combined into one force, g, acting perpendicular to the sail/rudder.

$$\begin{aligned} g &= L\cos(\alpha) + D\sin(\alpha) = \\ &= k\frac{1}{2}v_{\mathrm{a}}^2(\sin(2\alpha)\cos(\alpha) + (1 - \cos(2\alpha))\sin(\alpha)) = \\ &= k v_{\mathrm{a}}^2 \sin(\alpha) \end{aligned} \tag{2.7}$$

where $k = k_1\rho A$.

2.4.1 Sail force

The angle of attack, α_s, on the sail is determined by the direction of the apparent wind and the angle of the sail, $\alpha_s = \psi_{\mathrm{aw}} - \delta_s$, where the apparent wind is given by equation (2.4). The force on the sail is thus given by,

$$g_s = -p_4 a_{\mathrm{aw}} \sin(\delta_s - \psi_{\mathrm{aw}}) \tag{2.8}$$

where p_4 is the lift coefficient of the sail. When the velocity of the apparent wind increase, the roll angle of the boat will increase resulting in a smaller effective area

of sail. This is in parts accounted for by skipping the square on the apparent wind speed. The sheet is flexible and therefore the sail cannot hold against the wind and thus stall the boat. This is accounted for by letting the model update the angle of the sail with equation (2.9).

$$\delta_s = -\text{sgn}(\psi_{\text{aw}}))\min(|\pi - |\psi_{\text{aw}}||, |\delta_s|) \tag{2.9}$$

where $|\psi_{aw}| \leq \pi$ and sgn is the sign function.

2.4.2 Rudder force

The apparent velocity of the water near the rudder is assumed to be parallel to the boats heading, e.g. no currents or vortexes produced by the hull affects the rudder. The angle of attack then becomes, $\alpha_{\text{r}} = -\delta_{\text{r}}$, and the magnitude of the apparent velocity becomes equal to the boats velocity. The force on the rudder is thus given by,

$$g_r = -p_5 v^2 \sin(\delta_r) \tag{2.10}$$

where p_5 is the lift coefficient of the rudder.

To make the model in equation (2.1) easier to read the sail and rudder forces is redefined by moving the minus signs from equation (2.10) and (2.8) into (2.1). In summary, the sail and rudder force is given by,

$$\boldsymbol{g} = \begin{bmatrix} g_s \\ g_r \end{bmatrix} = \begin{bmatrix} p_4 a_{\text{aw}} \sin(\delta_s - \psi_{\text{aw}}) \\ p_5 v^2 \sin(\delta_r) \end{bmatrix} \tag{2.11}$$

2.4.3 Friction

The boat is affected by some frictional forces, first a tangential friction force acting in the opposite direction of the sailboats heading. This force is assumed to be proportional to the square of the sailboats speed, $p_2 v^2$. The model also includes a rotational friction force proportional to the sailboats speed and rotational speed, $p_3 \omega v$. The frictional forces could also be seen as damping elements in the model.

2.5 Model simulations

For simulations of the model, presentation of recorded data and comparisons of simulations and experiments the software MATLAB is used. The model is implemented in a m-file and could be called at as a function. When the model function

is given the current state of the sailboat, true wind and control signals for rudder and sail, the derivative of each sailboat state is returned. The function includes all of the equations presented in the chapter so far and the values of the model parameters from table 2.2. For simulations are the differential equations in the model solved with MATLAB's differential equation solver *ode45*. The solver is given a handle to the model function and a time interval to solve for, in this interval the wind and the control signals are assumed to be constant. Between each interval the values of the wind and control signals could be updated. For most simulations is the interval length set to 1 s, resulting in a controller running at 1 Hz.

The controllers and filters in chapter 3 and 4 are also implemented as functions with the function calls stated on the form in listing 2.1. With this structure it is easy to compare different parts, e.g. different controllers or filters, by making small changes in the main file. Also, as stated in section 1.6.2, blocks with MATLAB code could be created in the LabVIEW program used for collecting data, and thus limiting the risk of implementation errors when porting code from simulations to the platform used for experiments.

Listing 2.1: Functions calls in MATLAB for code written in this project.

```
function dstate_dt = model(t,state,W,u)
function [delta_r, delta_s] = controller_X(state, W, param)
function [xhat_out, P_out] = filter_X(T,measurments,Q,R)
```

2.6 Experiments

The goal of the first experiments was to collect data for model parametrization and for filtering. Later, experiments for testing a controller were performed. In order to collect good data for the model parametrization some test cases were chosen. Similar cases could be found in [3], where they are used in tuning and evaluation processes of ship autopilots.

- **Circle** The sailboat sails in circles by setting constant rudder angle and sheet length. This tests the sailboats ability to handle wind from all directions and its acceleration when the wind starts to fill the sails.
- **Straight line** The sailboat sails with wind coming from the side, neutral rudder and constant sheet length. This case tests the balance between the forces generated by the sails and friction, together with the size of the drift.
- **Tacking** The sailboat tacks, that is sails on alternating sides of the wind

and therefore advances towards the wind. This is the most complex case, testing the interaction between all parts of the model, especially rudder and sail forces.

In the experiments data from all sensors is sent to a laptop over the wireless link, and in real time presented to the user and stored for later analysis. The control signals to the actuators is decided by the user on the laptop and sent to the sailboat over the wireless link and also saved in the data files. Due to the design of the software on the laptop and on board the sailboat a delay of 1-3 seconds in the control signals is experienced. In this delay, also some of the time it takes to move the actuator is included, the rudder could be adjusted with about $16°/s$. Meaning that the rudder needs about 2 seconds to go from neutral to its maximal angle.

It is the relation between the force parameters (p_2, p_3, p_4 and p_5) and the mass/inertia that are important rather than values of the parameters them self. The parametrization is done by first estimating the sailboats mass, $p_9 = 300\,\mathrm{kg}$, and its moment of inertia. The moment of inertia is calculated with the corresponding equation for a rectangular plate with length l and width w,

$$p_{10} = p_9 \frac{l^2 + w^2}{12} = 400\,\mathrm{kg\,m^2} \tag{2.12}$$

where $l = 4\mathrm{m}$ and $w = 1\mathrm{m}$. The distance parameters (p_6, p_7 and p_8) is estimated from their physical interpretation on the boat. Since the boat has two sails and the model only has one it is impossible to measure the parameters directly.

All other model parameters, that could not be estimated by logical reasoning, were chosen in an iterative way by changing parameter values and comparing the trajectory of simulations and experiments. In the simulations measurements from corresponding experiment of wind, rudder angle and sail angle are used as input to the model. As a start parameters from [8] was chosen. Both the result from the parametrization and the values in [8] could be found in table 2.2.

2.6.1 GPS coordinates and great circles

The GPS unit gives the current position in decimal degrees relative the equator and the prime meridian. In the model it is assumed that the sailboat is moving in a plane, not an a sphere, and position is given as distances to the origin along the x- and y-axis, e.g. a local navigation frame is used. And also, the position output from simulations are given in meters relative the origin. In order to make

it possible to compare simulations and experiments it is necessary to map GPS coordinates to rectangular coordinates in the local navigation frame.

Since all experiments is conducted in the western harbour of Mariehamn, 60.1074° and 19.9218° was chosen as the latitude and longitude of the origin, e.g. the center of the local navigation frame. If both the coordinate of the origin and the present position is expressed in radians the distance from the origin is given by the spherical law of cosine,

$$\begin{aligned} \Delta\sigma_x &= \arccos(\sin(\phi_o)\sin(\phi_o) + \cos(\phi_o)\cos(\phi_o)\cos(|\lambda_o - \lambda_p|)) \\ \Delta\sigma_y &= \arccos(\sin(\phi_o)\sin(\phi_p) + \cos(\phi_o)\cos(\phi_p)\cos(|\lambda_o - \lambda_o|)) \\ s_x &= r\,\Delta\sigma_x\,\mathrm{sgn}(\lambda_p - \lambda_o) \\ s_y &= r\,\Delta\sigma_y\,\mathrm{sgn}(\phi_p - \phi_o) \end{aligned} \tag{2.13}$$

where ϕ_o, λ_o and ϕ_p, λ_p are the latitude and longitude of the origin and the position, sgn is the sign function and r is the radius of the earth[8]. The position in meters relative the origin is now given by $\begin{bmatrix} s_x & s_y \end{bmatrix}$.

2.7 Comparison of simulation and experiment

In figures 2.4–2.9 the trajectories and speed profiles from simulations and experiments described in section 2.6 are found. The blue arrow in the middle of the trajectory figures indicates the mean wind direction during each experiment. The, by the sailboat measured, mean true wind speed during each experiment was: 2.9, 3.5, 5.0 m/s for respective case. Figure 2.8 shows tacking manoeuvres, the rudder control signal from the experiment in this simulation is slightly modified. A glitch of ±3° is applied, within the glitch is δ_r set to zero. This removes small rudder activity that didn't affected the sailboat, but affected the model. It should be noted that the rudder actuator is connected to the rudder with ropes that has some elasticity. That is a likely explanation to why the sailboat didn't react to the small rudder adjustments made during this experiment.

In figure 2.4 and 2.5 it is seen that the simulated speed corresponds quite well to the measured speed but a bigger difference is noted in the trajectories. In the experiment the sailboat sailed a little bit straighter for a while, lower left corner in figure 2.4. The reason for this straighter trajectory is not clear, but it could explain parts of the difference of the trajectory and why the sailboat in the simulation turns to early.

[8]Earth radius, $r = 6371$ km

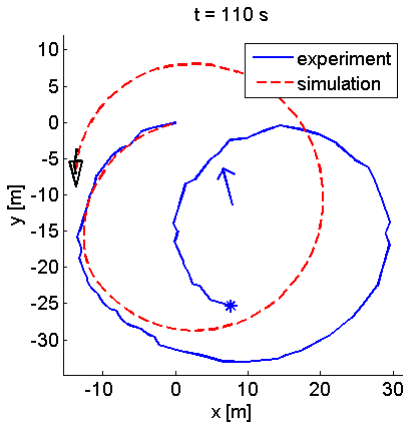

Figure 2.4: Test case Circle. GPS position from experiment and simulated trajectory using constant rudder resulting in circles.

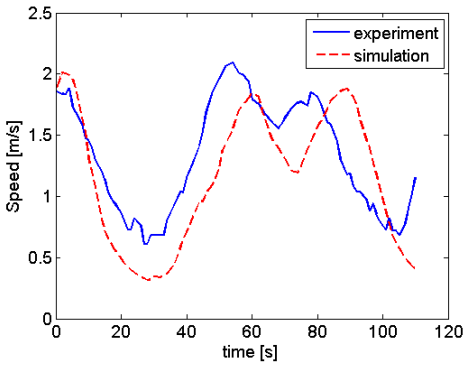

Figure 2.5: Test case Circle. Measured speed over ground from GPS in experiment and simulated speed as function of time.

In figure 2.6 and 2.7 the trajectory and speed profile of simulation and experiment are consistent.

In figure 2.8 and 2.9 again some differences in trajectories are seen. This test is complex and it is hard to simulate the correct heading after a couple of tacks. The integrated error in heading will rapidly be so large that no relevant comparison could be made if the parametrization of the model is bad.

The parametrization of the model is done on basis of the data in above mentioned figures. The parameter values chosen, used in all of the presented simulations, is presented in table 2.2. Most parameters have similar values as the original ones in [8], but some differences exists even though the two sailboats are of similar size, same weight. For example, the tangential friction is larger and the sail lift is smaller. A possible explanation could be that the sailboats has different rigs, the sailboat in [8] has a balanced rig instead of a sloop rig. Other noticeable differences is the moment of inertia and the rudder break coefficient, were the later doesn't exist in [8].

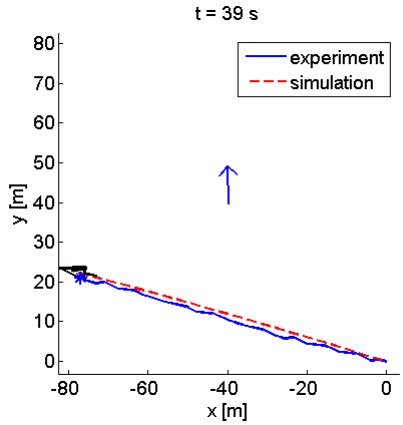

Figure 2.6: Test case Straight line. GPS position from experiment and simulated trajectory using neutral rudder resulting in a straight line.

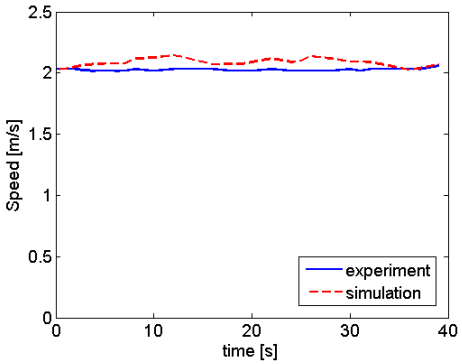

Figure 2.7: Test case Straight line. Measured speed over ground from GPS in experiment and simulated speed as function of time.

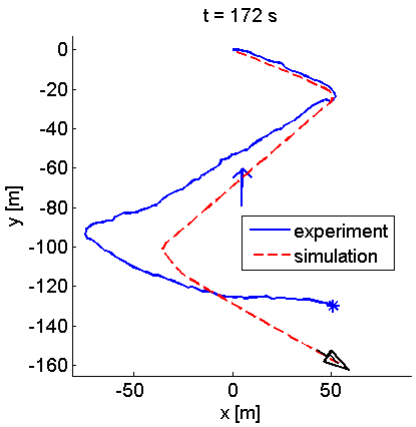

Figure 2.8: Test case Tack. GPS position from experiment and simulated trajectory.

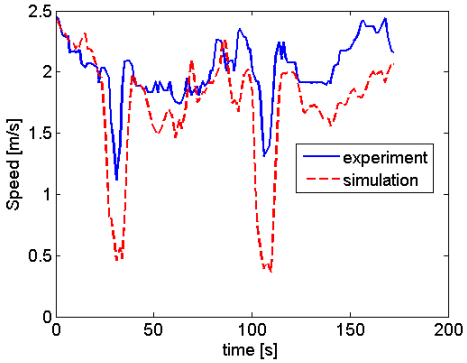

Figure 2.9: Test case Tack. Measured speed over ground from GPS in experiment and simulated speed as function of time.

Table 2.2: Model parameters and their values. To the right, the original values from [8].

	value	unit	explanation	original value
p_1	0.03	-	drift coefficient	0.05
p_2	40	kgs^{-1}	tangential friction	0.2
p_3	6000	kgm	angular friction	6000
p_4	200	kgs^{-1}	sail lift	1000
p_5	1500	kgs^{-1}	rudder lift	2000
p_6	0.5	m	distance to sail CoE	1
p_7	0.5	m	distance to mast	1
p_8	2	m	distance to rudder	2
p_9	300	kg	mass of boat	300
p_{10}	400	kgm^2	moment of inertia	10000
p_{11}	0.2	-	rudder break coefficient	-

Chapter 3

Control

Naturally, for a sailboat the propulsion is generated by sails only. This is an interesting characteristic since the wind from a traditional control perspective is defined as a disturbance. In this case, however, the disturbance is necessary since without any wind the boat will obviously come to rest, regardless of efforts made by any controller.

A controller is an object which typically takes references and measurements of the system as input and by comparing these it gives control signals as outputs. The control signals are used as input to the system and by closing the system in a feedback loop the system follows the reference signals decided by the user. A common controller is the PID-controller, where P stands for proportional, I for integral and D for derivative. PID-controllers exist in many forms, one possible implementation is given by,

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)\, d\tau + K_d \frac{d}{dt} e(t) \tag{3.1}$$

where $e(t) = r(t) - y(t)$ is the the control error, e.g. the difference between the reference $r(t)$ and the measurement $y(t)$, and K_p, K_i, K_d are parameters used to tune the controller to a desired behaviour.

A sailboat is a complex system and it is hard to define a simple controller that follows a reference, e.g. a line, under all conditions. One problem is how to define a suitable control error. For example all sailboats have a no-go angle, e.g. a section on both sides of the wind direction where it is impossible to sail as a sailboat can not sail straight up in the wind. If the desired heading is within the no-go angle, the sailboat must tack, and the desired heading has just changed to something that is not straight towards were we would like to sail. If tacking is not considered

an intuitive control error would be the difference between the actual heading θ and the desired heading θ_r, $e = \theta - \theta_r$. It is even harder to construct a control error for the sail, but it is not really necessary to do so, since the best sail angle primarily is determined by the apparent wind angle. And therefore no control error for the sail is defined here.

Most controllers for sailboats are of "cascade type", in the meaning that the controllers work in several levels and deals with different problems on each level. A common overall structure is the use of an outer control loop handling the generation of a reference, e.g. a desired heading. And an inner control loop taking the reference as input and together with measurements generating control signals to the actuators.

3.1 Control strategy

As a starting point for controller development, a line following controller inspired by [7] is used. As inputs to the controller, the state variables X, true wind, $\boldsymbol{W}_{p,\text{tw}}$, and a series of way-points, P, are used. The goal is to keep the boat along the (linear) trajectories connecting the way-points. The trajectories are assumed to be free of obstacles. Some variables for internal use by the controller are the tack variable, $q = \{-1, 1\}$ used to remember the direction of the ongoing tack and a counter k keeping track of current way-points. Also, the controller is provided with a tacking angle θ_t, which decides how far off from the true wind angle the sailboat should sail when tacking, and a no-go angle, $\theta_{\text{no-go}}$, that determines when it is necessary to tack. The tacking angle and the no-go angle is dependent on the boat type used. The best tacking angle for most sailboats is roughly around 40° to 60°. The no-go angle can be defined as a function of wind speed, but are typically between 30° and 50° on either side of the true wind, ψ_{tw}. In addition, two distance-related parameters are provided to the controller. The distance r, determines the size of the way-point, i.e., at what proximity is it considered that the way-point has been reached. The distance, d, determines how close the sailboat will keep to the desired trajectory during tacking. Three different ways of calculating the rudder angle is presented. The sheet angle, δ_s, is a linear function of the apparent wind angle, ψ_{aw}.

The control algorithm is as follows.

1. Calculate the distance to next way-point r_1. If $r_1 < r$, the way-point is reached and the way-point counter is updated, $k = k + 1$.

2. Calculate the desired heading θ_r based on the shortest (signed) distance, l, from the boat to the desired trajectory by

$$\theta_r = \beta - 2\frac{\gamma}{\pi}\arctan(\frac{l}{r}) \tag{3.2}$$

where β is the angle of the desired trajectory and $\gamma > 0$ is a tuning parameter, i.e. a larger value for γ gives a trajectory of the boat that converges faster to the desired line.

3. Determine mode of sailing, nominal or tack. In nominal mode, go to step 4. If tacking is required, that is, true wind lies within the no go zone, q is set to 1 or -1 depending on the direction of the tack and if the sailboat has reached the tacking distance d. The desired heading is correspondingly set: $\theta_r = \psi_{\text{tw}} + q\theta_t$.

4. Calculate rudder angle. Three different controllers are presented in section 3.1.1

5. Calculate sail angle (sheet length), which in all controllers are proportional to the angle of the apparent wind,

$$\delta_s = -\text{sgn}(\psi_{\text{aw}})\left(\frac{\delta_{s,\min} - \delta_{s,\max}}{\pi}|(\psi_{\text{aw}}| + \delta_{s,\max}\right). \tag{3.3}$$

The values of the control parameters in the outer control loop is presented in table 3.1.

Table 3.1: Parameters used in the controller

waypoint size	$r = 35\,\text{m}$	$\delta_{r,\max} = \pi/6\,\text{rad}$
incidence angle	$\gamma = \pi/4\,\text{rad}$	$\delta_{s,\min} = \pi/32\,\text{rad}$
tacking angle	$\theta_t = \pi/3\,\text{rad}$	$\delta_{s,\max} = \pi/5.2\,\text{rad}$

3.1.1 Rudder control

Step 4 in the control algorithm involves the rudder control, here three controls strategies for the rudder is presented. These will later be evaluated in simulations and the last is also tested in an experiment on the real sailboat. All of the controllers uses the same definition of the control error, $e = \theta - \theta_r$. To make sure that the sailboat always turns in the "best" direction e must be mapped to its equivalent in the interval $[-180^\circ\ 180^\circ]$ if e is outside of this interval. The controller

output equals the desired rudder angle, $u = \delta_r$. When implementing this on a real sailboat with an actuator is an other abstraction layer required, e.g. the desired rudder angle must be translated to control signal understood by the actuator. In the experimental setup used in this project the command sent to the actuator is a linear function of the rudder angle.

The first controller is a simple *P-controller*

$$u_{\mathrm{P}}(t) = K_p e(t). \tag{3.4}$$

The second controller is a *PD-controller* which is common in autopilots for ships. In order to implement the *PD-controller* in a discrete system the derivative in (3.1) is discretezied, resulting in,

$$u_{\mathrm{PD}}(k) = K_p e(k) + K_d \frac{e(k) - e(k-1)}{\Delta t} \tag{3.5}$$

As final step in the *P-* and *PD-controller* the output is "forced" to be within the interval $\pm\delta_{r,\max}$, where $\delta_{r,\max}$ is the maximal rudder angle. The third controller comes from [7] and could be described as a *sin-controller*, here are the control output proportional to sinus of the control error,

$$u_{\sin}(k) = \sin(e(k))\delta_{r,\max} \tag{3.6}$$

If the boat is going in the wrong direction, i.e. $\cos(e(k)) < 0$, maximal rudder angle is used,

$$u_{\sin}(k) = \mathrm{sgn}(\sin(e(k))\delta_{r,\max} \tag{3.7}$$

This controller reminds a lot of a P-controller with $K_p = \delta_{r,\max}$, the only difference is a smooth transition to the maximal rudder angle, as seen in figure 3.1. In the figure the rudder angle (control signal to the rudder) is plotted as a function of the control error for three different controllers.

3.2 Rudder control evaluation

In order to compare and evaluate performance for different controllers some form of measure is necessary. Four measures for control evaluation are defined. In the simulations each controller will be tested by letting the sailboat sail along the shape of a triangle, where one leg is against the wind and therefore tacking is necessary. Naturally, the time it takes to complete one lap is of interest, t_f. The rudder activity, e.g. the control signal activity is defined by $\sum |\delta_r|$. A smaller value of the rudder activity means that the rudder actuator is working less, resulting in

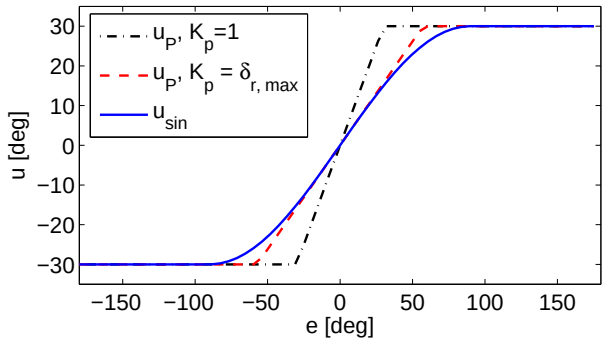

Figure 3.1: Comparison of rudder angle as a function of the control error in the region $[-180°\ 180°]$ for three different controllers. Proportional control ($K_p = 1$) dot-dashed (.-), proportional control ($K_p = \delta_{r,\max}$) dashed(– –) and *sin-controller* solid (—).

Table 3.2: Performance evaluation of rudder control

		t_f	$\sum \lvert\delta_r\rvert$	$\sum \lvert\theta - \theta_r\rvert$	$\sum \lvert l\rvert/1000$
P:	$K_p = 1$	520	72.3	139	6.48
PD:	$K_p = 1,\ K_d = 0.5$	521	71.7	141	6.60
P:	$K_p = \delta_{r,\max}$	528	63.4	151	6.67
sin-controller:		533	61.3	154	6.90
PD:	$K_p = \delta_{r,\max},\ K_d = 0.5$	535	61.8	157	6.82

lower power consumption. The control performance is defined by the control error and distance to the reference line, e.g. $\sum |e|$ and $\sum |l|$.

Five controllers for the rudder are evaluated. In table 3.2 the result for each controller is presented. The fastest lap time and best control performance is achieved by the P-controller with $K_p = 1$ with the expense of the highest control activity. The sin-controller has the smallest control activity and the PD-controller is somewhere between the two others in overall performance. All controllers were updated with 1 Hz. In figure 3.2 the trajectory of the sailboat is shown when the rudder controller is of P type. The overall shape of the trajectory only has small differences if any of the other controllers presented are simulated. No noise is added to the "measurements" of heading, e.g. the "true" value of the heading is used by the controller when constructing the control error. In a real experiment would a compass measure the heading and the measurement will include some noise. Depending on the characteristics and size of the noise could a filter on the derivative term in the *PD-controller* improve its performance.

In figure 3.4 the trajectories from an experiment of the whole system together with its corresponding simulation is found. At the time of the experiment no

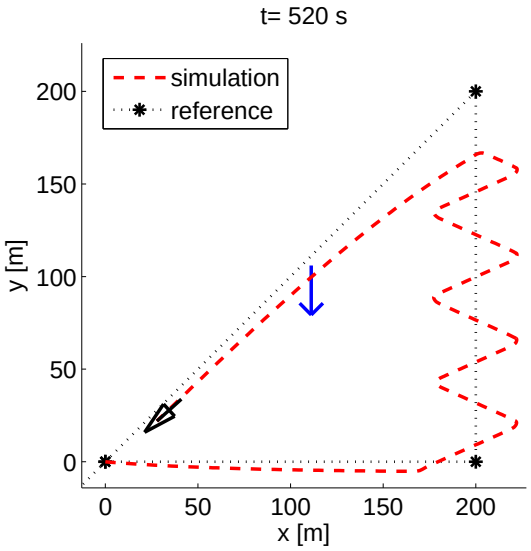

Figure 3.2: Simulation where the sailboats completes a triangle with a P controller acting on the rudder with $K_p = 1$, the simulated sailboat needs 520 s to complete one lap. A waypoint is considered reached when the sailboats is within 35 m from its center.

other controller than the *sin-controller* in equation 3.6 had been investigated. The controller in both the simulation and the experiment were given the same points to construct a reference line from. The controller will steer the boat along the reference line from the starting point towards the points 1-6. The reference points are chosen with regards to the shape and obstacles present in the area were the experiment were performed. The experiment lasted about 13 minutes, with a westerly wind of approximately 7 m/s.

In the figure, it can be seen that the sailboat in the simulation follows the reference nicely, but with statical control error downwind from the reference. The sailboat in experiment experienced big oscillations along the trajectory, especially in the beginning. A possible explanation for this observation is a varying time-delay due to, mainly, communication delays between the sailboat and the external laptop. If a time-delay is included in the model and simulated, similar oscillations can be observed. Because of the oscillations, the maximum rudder angle in the controller, $\delta_{r,\max}$ in equation 3.6, was changed from $\pi/6$ to $\pi/9$ after 300 seconds of the experiment. Although this clearly improved controller performance, the experiment shows the need for further improvement. Also, an obvious drawback of this modification is that maximum rudder can no longer be applied. The trial thus reveals the need to modify the "gain" of the controller without adjusting maximum rudder angle. In figure 3.3 the control signal before and after the change in maximum rudder angle is shown. It is clear from the figure that the gain de-

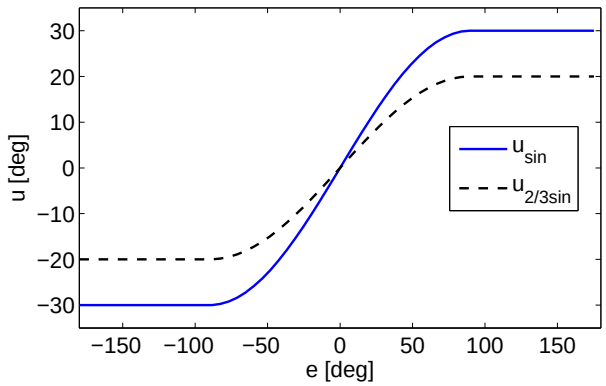

Figure 3.3: Comparison of rudder angle as a function of the control error in the region $[-180° \; 180°]$ for two *sin-controllers* with $\delta_{r,\max} = \pi/6$ solid (—) and $\delta_{r,\max} = \pi/9$ dashed (– –).

creased together with the maximum rudder angle, the gain could be seen as the slope of the graph in the region around 0°. It would be possible to manipulate the control error, e.g. scaling with a constant, and in that way change the gain without changing the maximum rudder angle.

At this point in the project the other controllers were investigated with simulations. Due to limitations in available time for experiments no experiments with the other controllers were performed.

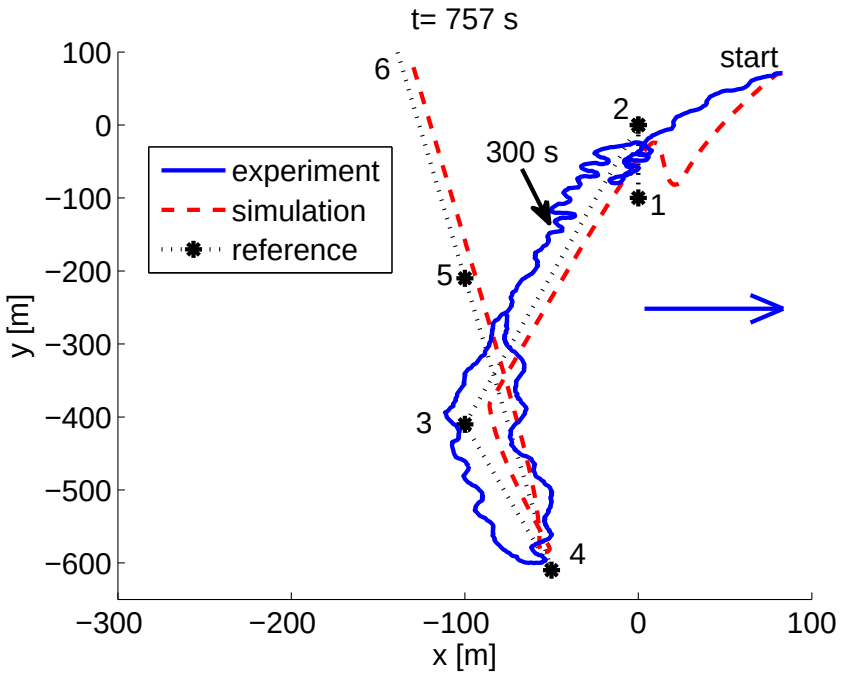

Figure 3.4: Experiment with sin-controller compared with simulation. Desired trajectory, dotted (..), simulated path, dashed (– –), and experiment solid (—).

Chapter 4

State estimation

A desired feature of the autonomous system is a very low power consumption during longer missions. As discussed in [1], this can be achieved by shutting down the measurement and control system for longer periods. Informed and safe choices to shut down and wake up the electronic system can be based on reliable estimates of the state of the boat, i.e., mainly the position. Given the form of the model and the promising simulations, this chapter gives a hint of the possibilities for state estimation with the use of Kalman filtering.

A Kalman filter is a method named after Rudolf E. Kálmán whom was highly involved in developing the theory. The aim of a Kalman filter is to improve measurements by filtering them with information about the underlying system producing the measurements and statistical information about noise included in the measurements. The Kalman filter also gives the possibility to get an update of the states without the need of new measurements. Since the model is non-linear an extended Kalman filter is used.

Kalman and extended Kalman filtering is widely used in many areas of control and signal processing. The following theory of this chapter is explained in greater detail, for example in [4, 5].

Two filters are introduced, one for filtering the measurements of apparent wind and a second filter is used for the system states. The filtered wind measurements are used as input to the sailboat system filter. The reason for using two separate filters is stability. It turns out to be much easier to get a stable filter if the wind model and the sailboat model are separated into two separate filters. It is not exactly clear why the combined filter tends to be unstable, some possibilities are discussed in chapter 5.

4.1 Sailboat filter

4.1.1 Discretization of model equations

The model as stated in equation 2.1 is given on continuous form. For filtering the model must be discretized, e.g. the system model needs to be expressed on discrete form.

$$\boldsymbol{x}_{k+1} = f(\boldsymbol{x}_k, \boldsymbol{u}_k) \tag{4.1}$$

where $\boldsymbol{x}$ is the state vector, $\boldsymbol{u}$ is the control signals, f is the non-linear discretized system equations and k marks the time step. Here the true wind parameters are considered to be part of the control signals, $\boldsymbol{u} = \begin{bmatrix}\delta_r & \delta_s & a_{\text{tw}} & \psi_{\text{tw}}\end{bmatrix}^{\text{T}}$.

The continuous derivative is approximated with

$$\frac{d\boldsymbol{x}}{dt} = \frac{\boldsymbol{x}_{k+1} - \boldsymbol{x}_k}{T} \tag{4.2}$$

where T is the system sampling time. Thus is the discretized system equations in equation 4.1,

$$f(\boldsymbol{x}_k, \boldsymbol{u}_k) = \begin{bmatrix} x_k + T(v_k \cos(\theta_k) + p_1 a_{\text{tw}} \cos(\psi_{\text{tw}})) \\ y_k + T(v_k \sin(\theta_k) + p_1 a_{\text{tw}} \sin(\psi_{\text{tw}})) \\ \theta_k + T\omega_k \\ v_k + T(g_s \sin(\delta_s) - g_r \sin(\delta_r) - p_2 v_k^2)/p_9 \\ \omega_k \end{bmatrix} \tag{4.3}$$

where the rudder and sail forces, g_r and g_s, are calculated in the same way as in the continues model. A process noise q_k will be added to all states. The process noise is assumed to be white zero mean Gaussian, with covariance matrix Q. Q is a static diagonal matrix and an important design parameter in the filter. It could be interpreted as an estimate of the model errors.

Take special notice of the difference in the fifth row in the discretized model compared to the continuous model. Due to stability problems the rotational speed equation is simplified to, $\omega_{k+1} = \omega_k + q_k$. The source of the instability is the damping term dependent of ω. The problem could be solved by decreasing the sampling/predict time or by simplifying the equation for ω_{k+1}. If the discretized model in equation (4.3) is simulated the model is only stable if the time step is small, more precisely $T < 0.04\,s$. In all filter implementations that follows the orientation θ is measured and there is no need of knowing ω, therefore the simplified version of ω_{k+1} is chosen. Similar problems could arise from the damping term in the equation for v_{k+1}, but this term is not as sensitive.

4.1.2 Measurement equations

The measurement equations, describing the relation between states and measurement, is trivial for the model states, e.g. a one-to-one relation.

$$\boldsymbol{y}_k = \begin{bmatrix} x \\ y \\ \theta \\ v \end{bmatrix} = h(\boldsymbol{x}_k) = \begin{bmatrix} x_k \\ y_k \\ \theta_k \\ v_k \end{bmatrix} \tag{4.4}$$

All measurements are assumed to include some noise r_k. The measurement noise is assumed to be white zero mean Gaussian, with covariance matrix,

$$R = \begin{bmatrix} \sigma_x^2 & 0 & 0 & 0 \\ 0 & \sigma_y^2 & 0 & 0 \\ 0 & 0 & \sigma_\theta^2 & 0 \\ 0 & 0 & 0 & \sigma_v^2 \end{bmatrix} \tag{4.5}$$

where σ_i is the standard deviation of measurement sensor i. It is the relation between R and Q that are important when designing the filter.

4.1.3 Extended Kalman filter equations

The whole non-linear system, with noise, system equations and measurement equations is now,

$$\boldsymbol{x}_{k+1} = f(\boldsymbol{x}_k, \boldsymbol{u}_k) + q_k \tag{4.6}$$

$$\boldsymbol{y}_k = h(\boldsymbol{x}_k) + r_k \tag{4.7}$$

The extended Kalman filter consists of two parts, a predict part and an update part. In the predict part the old states is used together with the model to get an estimate of what the states will be in next time step, equation 4.8. In the update part the measurements is used for estimating the states at the current time step, equation 4.14. The notation $k|k-1$ should be interpreted as the value at time k with measurements up to time $k-1$.

Predict equations

$$\hat{\boldsymbol{x}}_{k|k-1} = f(\hat{\boldsymbol{x}}_{k-1|k-1}, \boldsymbol{u}_{k-1}) \tag{4.8}$$

$$P_{k|k-1} = F_{k-1} P_{k-1|k-1} F_{k-1}^{\mathrm{T}} + Q \tag{4.9}$$

$$F_{k-1} = \left. \frac{\partial f(\boldsymbol{x}, \boldsymbol{u})}{\partial \boldsymbol{x}} \right|_{\hat{\boldsymbol{x}}_{k-1}, \boldsymbol{u}_{k-1}} \tag{4.10}$$

where P is the state covariance matrix, F^{T} is the transpose of F, and F is the systems Jacobian, e.g. the linearized system equation matrix.

Update equations
In the update part the measurements is taken into consideration by constructing the innovation, which is the differences between the real measurements and what the model predicts that the measurements should be, equation (4.11). The innovation covariance, equation (4.12) is then used to decide the Kalman gain, equation (4.13). It is now possible to update the state and its covariance matrix with equation (4.14) and (4.15).

$$\tilde{\boldsymbol{y}}_k = \boldsymbol{z}_k - h(\hat{\boldsymbol{x}}_{k|k-1}) \tag{4.11}$$
$$S_k = H_k P_{k|k-1} H_k^{\mathrm{T}} + R_k \tag{4.12}$$
$$K_k = P_{k|k-1} H_k^{\mathrm{T}} S_k^{-1} \tag{4.13}$$
$$\hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k|k-1} + K_k \tilde{\boldsymbol{y}}_k \tag{4.14}$$
$$P_{k|k} = (I - K_k H_k) P_{k|k-1} \tag{4.15}$$
$$H_k = \left. \frac{\partial h(\boldsymbol{x})}{\partial \boldsymbol{x}} \right|_{\hat{\boldsymbol{x}}_{k|k-1}} \tag{4.16}$$

where $\boldsymbol{z}_k$ is the latest measurements, S is the innovation covariance, K is the Kalman gain, I is the identity matrix and H is the measurement Jacobian.

When the innovation is calculated the values of the angles must be checked. On the unit circle is the difference, e.g. shortest distance, between two angles never larger then π, therefore is it some times necessary to correct the innovation angles. This is done by subtracting or adding 2π if the angles is larger or smaller then $\pm\pi$. This is the same thing as mapping the angles to its corresponding value in the interval $\left[-\pi\ \pi\right]$.

4.2 Wind filter

The wind filter is also implemented as a Kalman filter with the same filter equations as above, but the model and measurement equations differ. The true wind is assumed to slowly vary according to a random walk in discrete-time, i.e.,

$$\begin{bmatrix} \hat{a}_{\mathrm{tw}}(k+1) \\ \hat{\psi}_{\mathrm{tw}}(k+1) \end{bmatrix} = \begin{bmatrix} \hat{a}_{\mathrm{tw}}(k) + q_a \\ \hat{\psi}_{\mathrm{tw}}(k) + q_\psi \end{bmatrix} \tag{4.17}$$

where $\hat{a}$ is used to denote the filtered estimate of a and q is Gaussian noise. States for heading and sailboat speed are also added to the state vector. The wind model is linear and therefore the system is given by,

$$\begin{aligned} \boldsymbol{x}_{k+1} &= F\boldsymbol{x}_k + q_k \\ \boldsymbol{y}_k &= h(\boldsymbol{x}_k) + r_k \end{aligned} \tag{4.18}$$

where $\boldsymbol{x} = \begin{bmatrix} a_{\text{tw}} & \psi_{\text{tw}} & \theta & v \end{bmatrix}^{\text{T}}$, F is an identity matrix and $h(\boldsymbol{x}_k) = \begin{bmatrix} \boldsymbol{W}_{p,\text{aw}}^{\text{T}} & \theta & v \end{bmatrix}^{\text{T}}$.

True wind is assumed to slowly vary and therefore the corresponding process noise is set to a small value. Giving Q and R matrices as follows,

$$Q = \begin{bmatrix} 0.001 & 0 & 0 & 0 \\ 0 & 0.0001 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \; R = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0.1 \end{bmatrix} \tag{4.19}$$

4.3 Filter implementation and results

The sailboat filter and the wind filter were implemented in MATLAB. The wind filter has been tested and used in real time, but was executed on an external laptop, as previously described. For the implementation of the sailboat filter the equations in section 4.1.3 is used, even though it would be possible to use a toolbox, for example [5].

In figure 4.1 and 4.2 the filtered estimates of true wind can been seen together with true wind measurements. True wind measurements are calculated with data from the wind sensor, compass and GPS unit. The filtered data spans over 300 seconds and were collected during the controller experiment, same experiment as in figure 3.4. During this experiment the true wind was estimated from the wind filter developed here and used as input to the controller. Also for the same time and approximate location, the Finnish Meteorogical Institue provided a westerly wind direction ($0°$) and speed of $5 - 6$ m/s, which correspond quite well to the estimates.

The sailboat filter in section 4.1 has been implemented with measurements of position, orientation and boat speed. The output trajectory is almost identical to the measured, just as expected. If the sailboat filter is implemented only with measurement from the compass, a more interesting filter is obtained when talking about power savings. Since this filter is not using the GPS. It should be noted that the wind filter, that functions as input to the sailboat filter, needs the speed

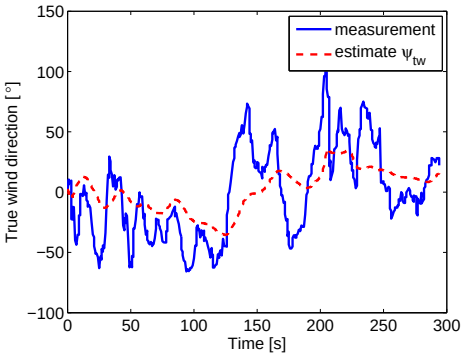

Figure 4.1: True wind direction, measured solid (—) and estimated dashed (– –).

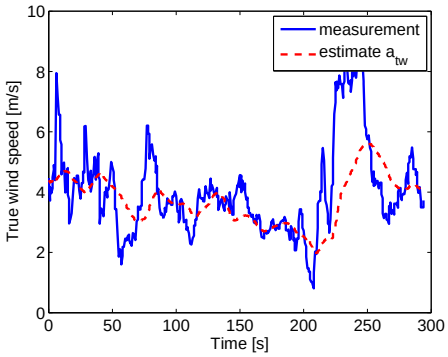

Figure 4.2: True wind speed, measured solid (—) and estimated dashed (– –).

measurement from the GPS. It should also be possible to estimate the sailboat speed from wind measurement, heading measurement and control signals. A trajectory comparison could be seen in figure 4.3. The estimated trajectory has very similar shape, logical since the orientation is measured, but the length of the trajectory is to short. In figure 4.4 is the same filter and data used, but the sail lift parameter in the model is increased from 200 to 500 $\mathrm{kg\,s^{-1}}$. When the sail lift parameter is increased the forces generated by the sail is bigger resulting in higher mean velocity. The speed will be to high if the larger sail lift parameter is used in experiments from section 2.6. This indicates that the parametrizations of the model is somewhat uncertain, more test cases and experiments are needed.

The sampling time in this experiment is varying. The system were sampled in 3 Hz, but some samples were lost resulting in mean sampling time of 0.4 seconds. With covariance matrix Q being diagonal of ones, and $R = 0.1$. The filter states are initialized with true values and state covariance matrix P as a square matrix of ones. The measured data spans over 730 seconds. This data set is not the most realistic since the boat oscillated around the reference line and therefore the rudder activity is unnecessary big. Also the constant and big change in heading is a problem. Even with this undesired characteristics of the data set the estimated path is quite good. With more experimental data, it might be possible to find better values of the model parameters. That in combination with a more realistic data set could possibly result in useful position estimates.

It would certainly be interesting to see what the result from an implementation of these filters would be if the filtering is done using different sampling rates. It would also interesting to see if the filtered states is good enough for control and most importantly if energy is saved. When talking about power savings it must be noted

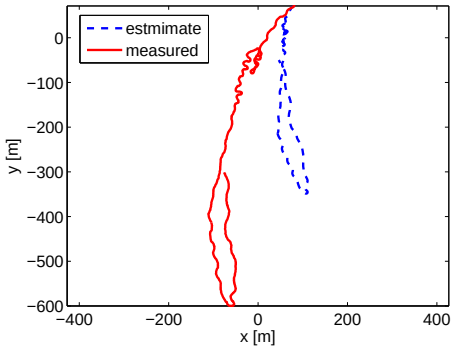

Figure 4.3: Trajectory from experiment solid (—) and sailboat filter output dashed (– –) when sail lift parameter set to its default value, $p_4 = 200\text{kg}\,\text{s}^{-1}$.

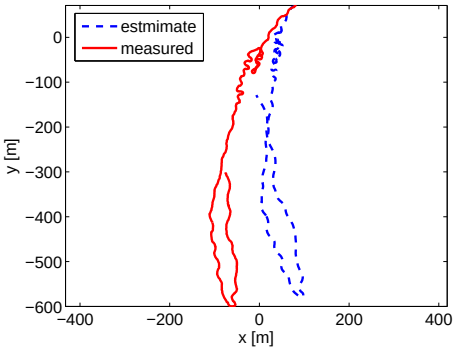

Figure 4.4: Trajectory from experiment solid (—) and sailboat filter output dashed (– –) with sail lift parameter increased, $p_4 = 500\text{kg}\,\text{s}^{-1}$.

that the power consumption of the electronic system in this setup is negligible in comparison with power used by the actuators. But the steering system could be replaced with something using less electrical power, e.g. a wind wane steering system.

Chapter 5

Concluding remarks

5.1 Conclusions and discussion

The developed model of the sailboat is rather simple with few parameters and five states, the model captures the main characteristics of a sailboat quite well, as is observed in experiments. It should also be said that there is room for better parametrization of the model, by the use of better data, longer data sets and more accurate measurements of rudder angle, and perhaps also an increased number of test cases.

The model seems to capture characteristics of the sailboat well enough for the purpose of controller design. Even though the controller that first was developed and tested with simulations showed bad behaviour when tested in an experiment. It is likely that the occurring oscillations would disappear if the controller is implemented exclusively in the on-board system since this would remove the time delay related to the wireless link. If the time delay is removed the decision of what rudder angle to use will be based on more resent data and therefore more accurate data. Also the time between a decision and actual movement of an actuator will decrease. A smaller time delay makes it easier to design a controller that steer the sailboat in a smooth way. Another way of dealing with the problem is to integrate a time delay in the model, and thus making it possible developing controllers capable of handling the time delays. This solutions is certainly not optimal since it deals with the effect of the problem rather than the source of the problem. And on a long mission is it absolutely necessary to have the controller implemented in the on-board system.

In order to further explore the non-linear features of the boat and compare con-

trollers, more experiments under different conditions are required. Such experiments, for example in different wind speeds and wave heights, would likely have an effect on the values of the model parameters. It could for example be obvious that some of the parameters are dependent on the wind speed and therefore better illustrating weaknesses in the model. Other experiment cases could also provided useful information, e.g. sailing with wind from behind. If new, longer experiments are performed conclusion about the models performance could be made with greater certainty. Especially the cases Circle and Straight line would benefit from longer data sets. 40 seconds seemed as a long time during the experiment, but when the data were analysed, and especially when talking about power savings this is a short time period, thus longer data sets are needed. The time periods that gives good results when shutting down and restarting a system/sensors for power saving purposes is likely in the regions of minutes or possible hours.

The parametrization was done by visually comparing the trajectories and speed profiles of experiments and simulations. This method is time consuming and not very accurate. If a cost function is introduced the parametrization process could be more effective and the result would likely be much better. Another benefit is that the result will be reproducible. The cost function could, for example, be a weighted sum of the squared error of the lateral trajectory and the speed profile.

The controller has not been tested during tacking with any experiments. The algorithm should work, since it has been tested in simulations, but experiments could reveal weaknesses in the algorithm or some parameters that needs tuning. It is not clear how well the algorithm performs when the wind is coming from an direction close to the no-go zone, and if the wind is alternatively inside and outside of the no-go zone.

In chapter 4, it is obvious that the wind measurement needs to be filtered, figure 4.1. The method used seems to produce good estimates of true wind, at least good enough for the control algorithm presented. Other, simpler methods could also provide a good estimate of true wind, e.g. mean filtering of true wind calculations.

The developed sailboat filter is close to an open-loop simulation of the model. It seems that a remaining challenge is to reliably estimate position as illustrated by the growing difference between observations and simulations in figure 4.4. Considering the endeavours connected to navigational satellite systems and the vast efforts under millennia of human civilization it is not, however, surprising that GPS measurements are vital for accurate position estimation. Still, it seems that wind and other state estimates can be used in open water to greatly decrease sampling rates and thus electrical power consumption. When a study of power savings

is performed with practical experiments the sate estimation filter should also be compared with estimates based on constant speed and heading.

As mentioned earlier the decision of making two filters were based on problems with stability in the filters. It is not exactly clear what the source of the problems are, but when the discretized model is simulated a relatively short time period between samples is needed, less than 0.04 seconds. Most of the time the problem starts in the equation for ω, especially the friction term which will be very big when ω is big. The friction term will also have alternating signs as the state goes to infinity. An other reason could be the way the discretization is done, other and perhaps more suitable methods exist. It is of course also possible that the problems solely lies within the implementation.

5.2 Further work

Recommendations for future work based on the findings of this thesis.

- Further experiments with the controllers integrated in the on-board system, in order to get an better understanding of the limitations of the model for controller development.
- Development of state estimation filters that can handle sensors with different frequencies. This kind of state estimation filters should then be used for an evaluation of the power saving strategy developed in [1].
- With an increased number of test cases and better data sets the parametrization and evaluation of the model should be of better quality and the results and usefulness of the model will likely increase.
- Improve the system, both software in the external laptop and in the on-board system, with the time-delays in mind. Some sort of interrupt based strategies could be applied. This would further increase the benefits of the wireless link and facilitate fast controller development.

Bibliography

[1] K. Dahl, A. Bengsén, and M. Waller. Power management strategies for an autonomous robotic sailboat. In *Robotic Sailing 2014: Proceedings of the 7th International Robotic Sailing Conference*, pages 47–56. Springer, 2014.

[2] Ronny Eriksson and Anna Friebe. Challenges for autonomous sailing robots. In *14th Conference on Computer and IT Applications in the Maritime Industries (COMPIT)*, pages 67–73, 2015.

[3] T.I. Fossen. *Marine Control Systems: Guidance, Navigation and Control of Ships, Rigs and Underwater Vehicles*. Marine Cybernetics, 2002.

[4] Fredrik Gustafsson. *Statistical sensor fusion*. Studentlitteratur, Lund, 2012.

[5] Jouni Hartikainena, Arno Solin, and Simo Särkkä. Optimal filtering with kalman filters and smoothers - a manual for the matlab toolbox ekf/ukf, 2011. http://becs.aalto.fi/en/research/bayes/ekfukf/documentation.pdf {2015-03-24}.

[6] Steffen Hessberger. Modeling and simulation of an autonomous sailing boat. Bachelor thesis, ETH Zürich, Switzerland, 2014.

[7] Luc Jaulin and Fabrice Le Bars. A simple controller for line following of sailboats. In *Robotic Sailing 2012: Proceedings of the 5th International Robotic Sailing Conference*, pages 117–129. Springer, 2012.

[8] Luc Jaulin and Fabrice Le Bars. Sailboat as a windmill. In *Robotic Sailing 2013: Proceedings of the 6th International Robotic Sailing Conference*, pages 81–92. Springer, 2013.

[9] David Krammer. Modeling and control of autonomous sailing boats. Master's thesis, ETH Zürich, Switzerland, 2014.

[10] Robin Lovelock. Team Joker, Snoopy's GPS Guided Trans-Atlantic Robot Boat, 2015. http://www.tsogpss.co.uk.gridhosted.co.uk/autop.htm {2015-04-21}.

[11] J. Melin, K. Dahl, and M. Waller. Modeling and control for an autonomous sailboat: A case study. In *Robotic Sailing 2015: Proceedings of the 8th International Robotic Sailing Conference*. Springer, 2015.

[12] Microtransat organisation. The microtransat challenge, 2015. http://www.microtransat.org {2015-06-21}.

[13] Roland Stelzer. *Autonomous Sailboat Navigation, Novel Algorithms and Experimental Demonstration*. PhD thesis, De Montfort University, Leicester, United Kingdom, 2012.

[14] Roland Stelzer and Karim Jafarmadar. History and recent developments in robotic sailing. In *Robotic Sailing: Proceedings of the 4th International Robotic Sailing Conference*, pages 3–23. Springer, 2011.

[15] Lin Xiao and Jerome Jouffroy. Modeling and nonlinear heading control of sailing yachts. *IEEE Journal of Oceanic Engineering*, 39(2):256–268, 2014.