NTNU
Norwegian University of
Science and Technology

Autonomous Sailboats

Modeling, Simulation, Control

Kristian Løken Wille

Master of Science in Engineering and ICT
Submission date: June 2016
Supervisor: Vahid Hassani, IMT

Norwegian University of Science and Technology
Department of Marine Technology

NTNU Trondheim
Norwegian University of Science and Technology
Department of Marine Technology

MASTER THESIS IN MARINE CYBERNETICS

SPRING 2016

for

KRISTIAN L. WILLE

Autonomous sailboats

Modeling, Simulation, Control

Work Description

The main objective of this thesis is to increase effectiveness and robustness of autonomous sailboats. Robustness is an important aspect of autonomy, but previous work have done little to address fundamental issues such as course control, reducing the danger of capsizing at high wind speeds, and how to optimally chose between different turning maneuvers when sailing up-wind.

A mathematical model of the system will be developed. The model will try to improve upon previous work by adding increased detailed, such as wind simulation and dynamics of the boom. The simulator will be used to verify the proposed control strategy.

Acknowledgements
I would like to express my gratitude to my supervisor Vahid Hassani for many great conversations and helpful comments on my work. I would also like to thank NTNU for five great years and a superb education.

Summary - English/Engelsk

This thesis is divided into five parts. In the first part, a general mathematical model of a sailboat is created. The model presented builds upon previous work and improves the modeling at multiple different areas such as current, wind, the boom dynamics, and more. Simulations are then executed and the results analysed to be used in the controller design.

In the next part, an ideal sail and rudder controller is designed on the basis that the system and its states are known at all times. The increased detailed in the simulation compared to previous work presents new challenges, such as how to control the angle of the sail when one only can do so by manipulating the length of a rope connected to the boom. New strategies for a course controller and a technique that reduces roll motion by using the sail are also presented and tested.

Next, a new and improved path following algorithm is created. Compared to previous works the solution presented is both easy to implement and provides good results. The path following algorithm also takes into account how to chose between tacking or jibing when sailing up-wind, increasing both performance and robustness.

The last two parts are about sensor simulation and state estimation, and how a practical control system can be designed. The practical controller do not have perfect state estimation and is designed to only account for the most important system dynamics, which are determined by using the analysis from the first and second part of this thesis.

Testing shows that every part of the system works as intended. The simulator increases the level of detail and captures new and interesting dynamics. The control system manages to hold a steady course and keep heeling angles within safe levels. The path following allows the sailboat to sail up-wind, and do so without risk of loosing control. The performance of the simple controller is similar to the ideal controller, despite being simpler.

Sammendrag - Norwegian/Norsk

Denne master oppgaven er delt inn i fem deler. Den første delen tar for seg matematisk modelering av en seilbåt. Denne modellen bygger på tidligere arbeid og utvider modellen ved å inkludere strømninger, forbedret vind modell, dynamikken til bommen, og mer. Simuleringer har blitt gjennomført og resultatene analysert.

Den neste delen tar for seg en perfekt seil og ror kontroll som er designet med tanke på at alt ved systemet er kjent til enhver tid. Den forbedrete modellen av seilbåten gjør at nye dynamikker ved systemet kommer frem, og det gir nye utfordringer som må løses, slik som å kontrollere seilet når man bare kan manipulere ett tau som er festet til bommen. Nye strategier for kurskontroll og en metode for å redusere rull bevegelse ved å bruke seilet har også blitt designet og testet.

Neste del tar for seg utviklingen av en algoritme som lar båten følge en gitt sti. Sammenlignet med tidligere arbeid er løsningen enkel å implementere og gir gode resultater. Algoritmen tar også for seg hvordan man best skal velge mellom forskjellige manøvere for å snu båten når man seiler opp mot vinden, noe som øker både effektivitet og robusthet.

De to siste delene handler om å estimere posisjon og orientering av skipet, og hvordan et mer praktisk kontroll system kan bli designet. Den praktiske kontrolleren antar ikke perfekt kunnskap om systemet og tar bare for seg de viktigste dynamikkene til systemet, som er blitt funnet ved hjelp av analysen i del en og to. Den praktiske kontrolleren yter veldig likt som den "perfekte" kontrolleren, selv om den er vesentlig enklere.

Testing viser at alle delene fungerer slik som de skal. Simulatoren oppfører seg mer realistisk en før og fanger nye og spennende dynamikker. Kontroll systemet klarer å holde båten på en stødig kurs samtidig som at båten ikke krenger for mye. Algoritmen for å følge en sti gjør at seilbåten kan seile mot vinden, uten å riskiere å miste kontroll.

Table of contents

Figures

List of Tables

1 Introduction

1.1 Motivation and Goals

The history of sailboats goes back to one of the earliest civilizations of mankind in Mesopotamia where the Tigris and Euphrates rivers inspired the development of many watercraft, including sailboats, see Carter [2012]. However, autonomy in sailboats is a topic that has gained attention only in recent years, see Xiao and Jouffroy [2014].

Sailboats require little to no energy to operate, making them well suited for long operations such as oceanographic research. Moreover, with the increased focus on clean energy, sailboats could provide a green option for transporting goods (Michael et al. [2014]). The propulsion force of the sailboat is created by the wind, and the only energy needed to operate the boat is by trimming the sail and controlling the rudder. Fitting the sailboat with solar panels or extracting energy directly from the sail itself makes them very much self sustainable in terms of energy (Jaulin and Bars [2013b]). Besides being cheap to operate, sailboats are fairly cheep to build and are reduces noise pollution as well as greenhouse gases.

The major downside of sailboats are their dependency on the weather conditions. The large surface area of the sail provides the boat with forward thrust, but it also makes the boat vulnerable against strong winds. This can cause large heeling angles, a lot of roll motion due to wind gusts, and in worst case it can cause the boat to capsize. Further more, the wind causes large side forces to be exerted on the boat causing relatively big drift angles compared to what is normally observer on motorized boats. Sailboats can not sail directly up-wind either, and one have to move in a zig-zag pattern to be able to sail up-wind.

The first part of this thesis will cover sailing basics; explaining the purpose of the sail, keel, rudder and some important maneuvers such as tacking and jibing. When this is established a mathematical model of a sailboat will be created, which will be used for simulation purposes and when designing the control system. Path following will allow the sailboat to sail up-wind, as well as increasing the efficiency when sailing down-wind. Lastly, sensor simulation and state estimation is added, and then a practical control system is designed by using all of the knowledge gathered from the previous chapters.

A lot of earlier papers on the topic of autonomous sailboats ignore a lot of the system dynamics, or simplify them to only the bear minimum is left, thus leading to very general PID controller solutions that has to make up for the model inaccuracies by relying heavily on integral action, trading simplicity for performance. In parts of this thesis the problem will be tackled from the other end to give a new perspective on the subject by trying to make an ideal controller, not taking into account restriction such as a non-perfect state estimation or model inaccuracies. It will help us understand which aspects of the system that are the most important, which will be useful when creating a more robust and practical controller later on that do not compromise on performance.

A general theme in this thesis is to increase the robustness of the autonomous sailboat. Sailboats will never be a viable option if they cannot handle a variety of different weather conditions, and robustness is an important aspect of autonomy. While the focus is on autonomous sailboats the automated control system could also help improve the sailing experience for a crew. A way of reducing roll motion by controlling the sail will be studied, that can improve both safety and comfort.

Two separate papers based on the work done in this thesis have been accepted for publication (Wille et al. [2016a,b]). The first paper is about modeling and course control, and the second paper is about roll stabilization.

1.2 Simplifications and Important Remarks

Even though the model presented in this thesis is more comprehensive than previous work it is still a very simplified model. The model of the sailboat presented do only include the mainsail, the keel and the rudder. The only way of controlling the sail will be through changing its angle relative to the wind, and it should be mentioned that there are numerous other ways of controlling a sail other than just changing the angle of it. For example, one can change the twist, the vertical camber, the draft and much more (C.A.Marchaj [2000]). Other types of sails can also be used together with the mainsail, such as a jib, genoa, or a spinnaker.

The following assumptions and simplifications are used throughout the thesis [1]:

- The yacht is rigid, and movement in heave and pitch are neglected.
- Waves, and the effect caused by them (first order and second order), are not modeled.
- Added mass coefficients are modeled as constants, in reality they are frequency dependent.
- The effects from the sail, keel, rudder and hull are computed independently of each other. As a result, effects caused by interactions between two (or more) of these are not modeled.
- The sail, keel and rudder are modeled as (rigid) foils. The lift and drag forces of a foil are due to the integration of pressure around the foil, though for simplification, the resultant force will be applied to a single point on the foil known as the center of effort.
- When modeling the boom dynamics the body-frame will be treated as inertial, and the movement of the boom will not effect the boat besides changing the angle of the sail.

Current is implemented in the mathematical model of the system, but the controller design do not take the current into account. This was partly to reduce the scope of the thesis, but it was decided to keep it in the model such that other may explore it in the future.

[1] The assumptions used are borrowed from Xiao and Jouffroy [2014], see source for more detail.

2 Basic sailing principles

2.1 Main Components

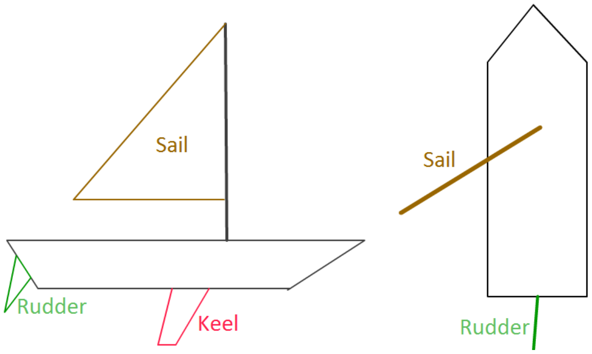


Figure 2.1: Main components of a sailboat shown from the side and from a top down view.

A sailboat has three main components: The sail, the rudder and the keel. The main task of the sail is to provide forward motion. Contrary to what one may intuitively think, the maximum speed is not reached when sailing directly down-wind, called running. When running the sail is limited to only using drag, and when the speed of the boat increases the effective wind speed on the sail decreases. When sailing perpendicular to the wind direction, known as beam reaching, the lift generated by the sail can be utilized, and when the speed of the ship increases the effective wind speed increases as well.

The rudder is used for steering the ship onto the desired course, just like on a motorized boat. However, on a motorized boat the rudder is usually placed behind the propeller, ensuring that the rudder always can provide some steering capabilities. On a sailboat one do not have this guarantee, meaning that some speed in surge is required to keep the boat controllable.

The keel is used to balance out unwanted forces in sway created (mainly) by the sail. The purpose of the sail is to provide forward motion, but when beam reaching the sail is also pushing the boat sideways. The keel creates an equal but opposite force in sway to keep the sailboat moving forward instead of sideways.

2.2 Tacking Maneuvers

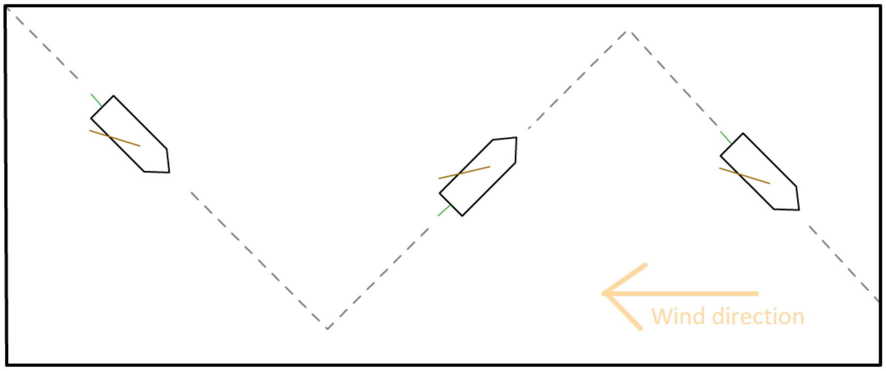


Figure 2.2: A sailboat that is beating up-wind.

A sailboat can not sail directly against the wind direction. Figure 2.2 shows the path the sailboat has to make to be able to sail up-wind. When sailing up-wind in this zig-zag pattern it is called beating, and the boat is said to be sailing close hauled when sailing close up against the wind direction. When beating the sailboat has to do multiple turns, and thees can be executed by doing a "tack" or a "jibe" maneuver. It is usually favorable to be sailing broad reach when sailing down-wind, which is at an angle between running and beam reaching, in a similar zig-zag pattern as when beating. When broad reaching one can utilize the lift from the sail, which as discussed earlier is more efficient.

Figure 2.3 shows the tack maneuver while beating. At one point the bow of the ship will be facing directly against the wind, also known as the eye of the wind, which is the defining characteristic of the tack maneuver. No forward forces are provided by the sail when tacking, and the ship will decelerate throughout the turn. If the speed falls too low the rudder will not be able to steer the boat.

Figure 2.4 shows the jibe maneuver when sailing up-wind. The jibe is defined by turning the stern of the ship through the eye of the wind. The main advantage of this compared to the tacking maneuver is that the sail will be providing a forward force to the boat. There are two disadvantages to jibing compared to tacking, the

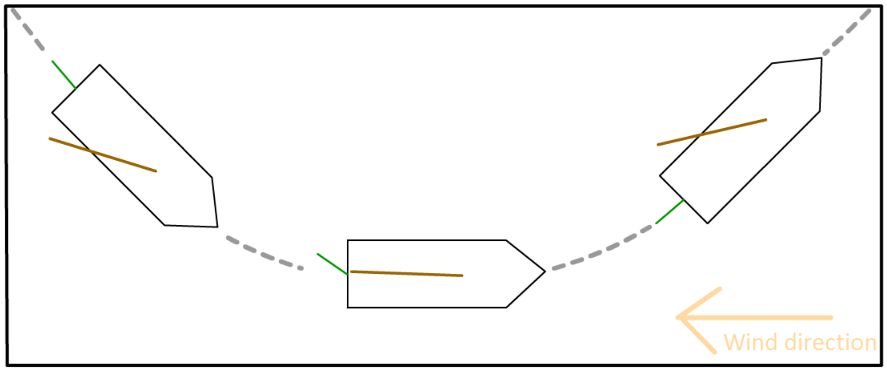


Figure 2.3: Detailed look at the tack maneuver

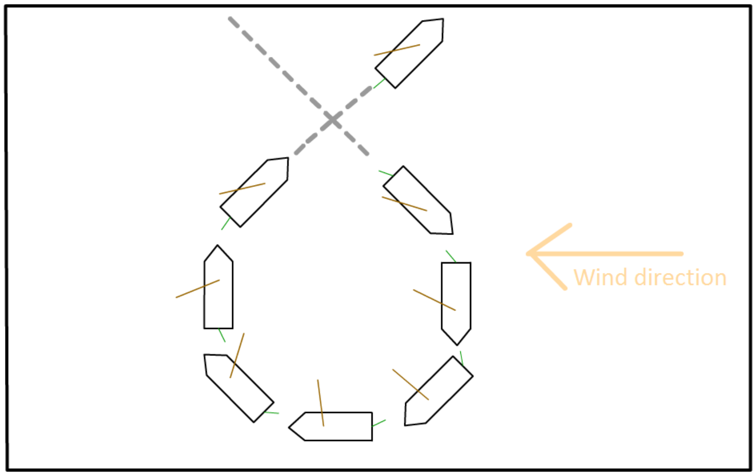


Figure 2.4: Detailed look at the jibing maneuver

first one being that it takes a longer amount of time when beating. The second problem is that the boom will swing over the deck at one point (as can be seen in figure 2.4), which can hit people if they are not prepared for it, or in worst case cause the boat to capsize due to the momentum of the boom.

3 System equations

Having a mathematical model that describes the dynamics of the sailboat is necessary before one should begin to discuss control. This is because optimal control theory is tightly connected to the system dynamics, and because it enables us to prove the effectiveness of our controller. The model developed will also be used for simulation and testing purposes, further increasing the need for a detailed model.

This chapter will be divided into three main parts. First, a mathematical model of the system dynamics are developed. Secondly, coefficients used in the system model will be estimated based on a real sailboat. The last part shows a couple of simulation results, and a discussion about which forces that dominates the system in different conditions.

The model presented builds upon the model developed by Xiao and Jouffroy [2014], which is the most advance sailboat model the author could find. Their model uses the framework developed by Fossen [2011] for modeling vehicles at sea, and adds detailed modeling of the sail, keel and rudder. However, there are some short comings in their model that will be addressed in this thesis, such as very simple drag model, no attention to modeling of actuators (sail, rudder), rudimentary wind modeling, and effects due to current are not included.

Main contributions of chapter:

- Realistic wind modeling.
- Accurate simulation of the boom.
- Developed model that includes effects due to current.
- Increased accuracy of drag modeling.
- Analysis of which forces that dominates during running and beam reaching.

3.1 System equations

3.1.1 Definition of forces

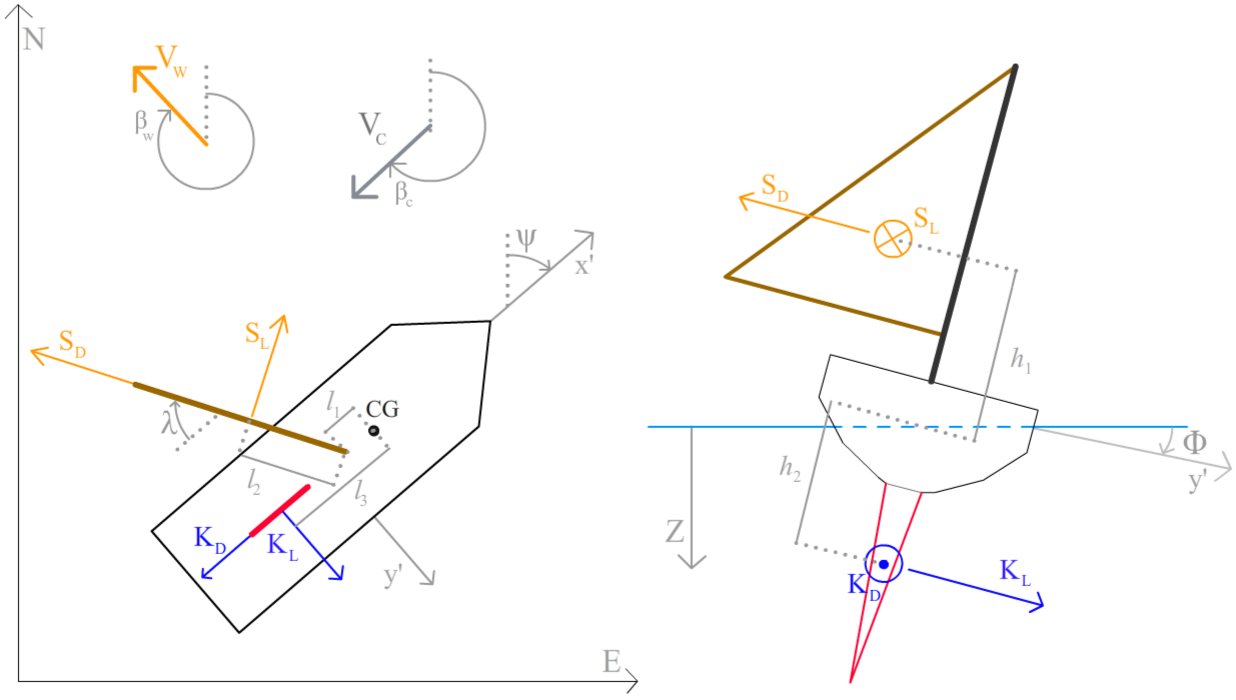


Figure 3.1: Definition of positive direction of important parameters, including forces created by the sail and the keel

Before presenting the system dynamics, description of the essential parameters in modeling the system, the direction of forces, axis and angles are presented in Figure 3.1 and 3.2. Explanation of variables can be found in table 3.1 and 3.2. All lift and drag forces are shown for an angle of attack equal to zero. Throughout the thesis, the notation of Society of Naval Architects & Marine Engineers has been adopted.

Variable	variable explanation
β_w	wind angle [rad]
V_w	wind vector $[\frac{m}{s}]$
β_c	current angle [rad]
V_c	current vector $[\frac{m}{s}]$
ψ	ship heading [rad]
ϕ	roll angle [rad]
λ	angle of sail [rad]

Variable	variable explanation
S_L	sail lift [N & mN]
S_D	sail drag [N & mN]
K_L	keel lift [N & mN]
K_D	keel drag [N & mN]
CG	Center of gravity [m]
x’	x in body frame [m]
y’	y in body frame [m]

Table 3.1: Variable explanation

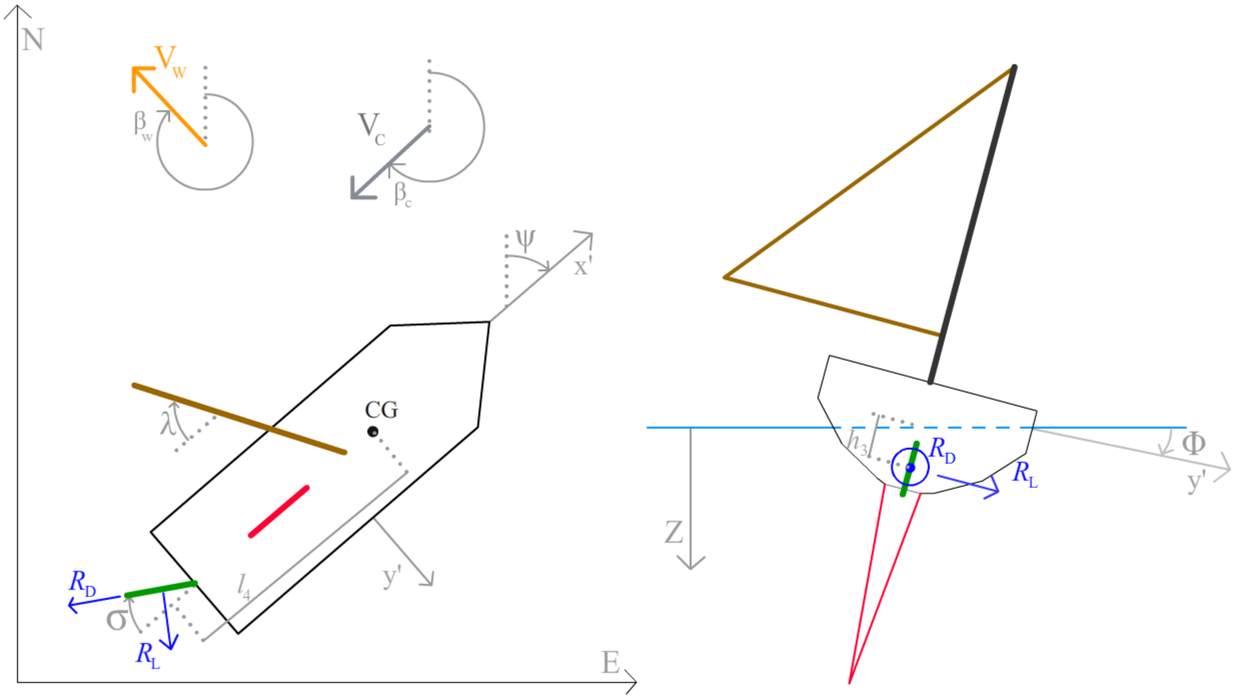


Figure 3.2: Definition of positive direction of forces created by the rudder.

Variable	variable explanation
σ	angle of rudder [rad]
R_L	rudder lift [N & mN]
R_D	rudder drag [N & mN]

Table 3.2: Variable explanation cont.

3.1.2 Vectorial representation

In what follows two different reference frames are used: the north-east-down (NED) reference frame (n-frame) and the body reference frame (b-frame). The NED coordinate system will be treated as inertial, and the b-frame is connected to the body of the ship. The origin of the b-frame, CO, is set to be at the center of gravity (CG) midship and at the waterline.

The equations of the system are based on the vectorial representation proposed by Fossen [2011]. A 4DOF[1] model has been chosen because of the big roll motions observed on sailboats. The states of the system become the following:

$$\eta = \begin{bmatrix} x \\ y \\ \phi \\ \psi \end{bmatrix} \qquad \nu = \begin{bmatrix} u \\ v \\ p \\ r \end{bmatrix}. \tag{3.1}$$

Variable	variable explanation
x	position in north [m]
y	position in east [m]
ϕ	roll [rad]
ψ	yaw [rad]

Variable	variable explanation
u	speed in x' direction $[\frac{m}{s}]$
v	speed in v' direction $[\frac{m}{s}]$
p	angular velocity in roll $[\frac{rad}{s}]$
r	angular velocity in yaw $[\frac{rad}{s}]$

Table 3.3: η and ν

The vectorial representation of the system is

$$\begin{aligned} \dot{\eta} &= J(\phi,\psi)\nu \\ M_{RB}\dot{\nu} + C_{RB}(\nu)\nu + M_A\dot{\nu_r} + C_A(\nu_r)\nu_r + D(\nu_r) + g(\eta) &= \\ S(\eta,\nu,\lambda,V_w) + K(\eta,\nu,V_c) + R(\eta,\nu,V_c,\sigma)&, \end{aligned} \tag{3.2}$$

where $J(\phi, \psi)$ is the transformation matrix

$$J(\phi,\psi) = \begin{bmatrix} cos(\psi) & -sin(\psi)cos(\phi) & 0 & 0 \\ sin(\psi) & cos(\psi)cos(\phi) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & cos(\phi) \end{bmatrix}, \tag{3.3}$$

[1] Degrees Of Freedom

Variable	variable explanation
M_{RB}, M_A	rigid/added mass matrix
$C_{RB}(\nu), C_A(\nu_r)$	rigid/added mass centripetal matrix
$D(\nu_r)$	drag matrix
$g(\eta)$	restoration forces
$J(\theta, \psi)$	transformation matrix from b- to n-frame
S	forces from sail
K	forces from keel
R	forces from rudder

Table 3.4: Variable explanation cont.

and $\nu_r = [u_r \ v_r \ p \ r]^T$ is the speed of the ship relative to the water

$$\begin{bmatrix} u_r \\ v_r \end{bmatrix} = \begin{bmatrix} u \\ v \end{bmatrix} - J_{2D}^T(\psi, \phi)V_c, \tag{3.4}$$

where J_{2D} is

$$J_{2D}(\phi, \psi) = \begin{bmatrix} cos(\psi) & -sin(\psi)cos(\phi) \\ sin(\psi) & cos(\psi)cos(\phi) \end{bmatrix}. \tag{3.5}$$

Assuming $\dot{V_c} = 0$, it follows that $\dot{\nu_r} = \dot{\nu}$.

M_{RB}, $C_{RB}(\nu)$, M_A, $C_A(\nu_r)$, $D(\nu_r)$, $g(\nu)$, $S(\eta, \nu, \lambda, V_w)$, $K(\eta, \nu, V_c)$ and $R(\eta, \nu, V_c, \sigma)$ are described in table 3.4. $S(\eta, \nu, \lambda, V_w)$, $K(\eta, \nu, V_c)$ and $R(\eta, \nu, V_c, \sigma)$ are dependent on several different states of the system, and will from now on be written as S, K and R for readability.

3.1.3 Mass and Centripetal Matrix

M_{RB} and M_A is expressed as (3.6) and (3.7). Added mass is a function of the frequency of the given motion, but for now, fixed added mass is used. The fixed added mass is chosen for a frequency of zero, except for the surge motion where it is chosen for an infinitely high frequency.[2]

[2] Only the most important cross terms are included.

$$M_{RB} = \begin{bmatrix} m & 0 & 0 & 0 \\ 0 & m & 0 & 0 \\ 0 & 0 & I_{xx} & 0 \\ 0 & 0 & 0 & I_{zz} \end{bmatrix} \tag{3.6}$$

$$M_A = -\begin{bmatrix} X_{\dot{u}}(\infty) & 0 & 0 & 0 \\ 0 & Y_{\dot{v}}(0) & 0 & Y_{\dot{r}}(0) \\ 0 & 0 & K_{\dot{p}}(0) & 0 \\ 0 & Y_{\dot{r}}(0) & 0 & N_{\dot{r}}(0) \end{bmatrix} \tag{3.7}$$

The rigid body coriolis matrix is equal to

$$C_{RB}(\nu) = \begin{bmatrix} 0 & -mr & 0 & 0 \\ mr & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.8}$$

and the added mass coriolis matrix, which causes the destabilizing Munk moment, $(Y_{\dot{v}} - X_{\dot{u}})u_r v_r$, is equal to

$$C_A(\nu_r) = \begin{bmatrix} 0 & 0 & 0 & Y_{\dot{v}}v_r + Y_{\dot{r}}r \\ 0 & 0 & 0 & -X_{\dot{u}}u_r \\ 0 & 0 & 0 & 0 \\ -Y_{\dot{v}}v_r - Y_{\dot{r}}r & X_{\dot{u}}u_r & 0 & 0 \end{bmatrix}. \tag{3.9}$$

3.1.4 Restoration forces

The restoring forces are calculated by

$$g = \begin{bmatrix} 0 \\ 0 \\ p_w g \Delta GM_t sin(\phi)cos(\phi) \\ 0 \end{bmatrix}, \tag{3.10}$$

where Δ is the total displacement of the ship and GM_t is the transverse metacentric height (Fossen [2011], p. 64).

3.1.5 Dampening

The damping matrix, $D(\nu_r)$, represents the damping caused by the hull of the ship (not the rudder and keel, which will be handled separately later). The damping matrix consists of a linear and a non-linear part:

$$D(\nu_r) = D_l\nu_r + D_q(\nu_r)\cdot \tag{3.11}$$

The linear damping is caused by potential damping and skin friction due to a laminar boundary layer, though only the latter will be considered. The potential damping is frequency dependent, and the effect is negligible at low frequencies.

$$D_l = \begin{bmatrix} d_{l_{11}} & 0 & 0 & 0 \\ 0 & d_{l_{22}} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & d_{l_{66}} \end{bmatrix} \tag{3.12}$$

Nonlinear damping is caused by wave drift and vortex shedding. Waves are not modeled, and follows that wave drift damping are not modeled either.

$$D_q(\nu_r) = [D_{q_X}(\nu_r)\ D_{q_Y}(\nu_r)\ D_{q_K}(\nu_r)\ D_{q_N}(\nu_r)]^T \tag{3.13}$$

The nonlinear damping in surge is based on the flat plate friction (ITTC 1957, Lewis [1988])

$$D_{q_X}(\nu_r) = \frac{1}{2} p_w S_H (1+k) C_F(R_n) |u_r| u_r, \tag{3.14}$$

where

$$C_F(R_n) = \frac{0.075}{(log_{10}(R_n) - 2)^2}, \tag{3.15}$$

p_w is the density of sea water, S_H is the wetted surface area of the ship, k is the form factor coefficient, $C_{DV}(R_n)$ is the friction coefficient due to nonlinear viscus effects and R_n is the Reynolds number. The nonlinear damping in sway and yaw is modeled using the cross-flow drag principle based on strip theory (Faltinsen [1990]):

$$D_{q_Y}(\nu_r) = \frac{1}{2} p_w \int_L C_D(x') T(x') |v_r + x'r| (v_r + x'r) dx' \tag{3.16}$$

$$D_{q_N}(\nu_r) = \frac{1}{2} p_w \int_L C_D(x') T(x') x' |v_r + x'r| (v_r + x'r) dx', \tag{3.17}$$

where $T(x')$ is the draft and $C_D(x')$ is the two-dimensional drag coefficient. The roll damping is modeled using the theory developed by Himeno [1981]. The roll damping consists of a friction and a lift damping component

$$D_{q_K}(\nu_r) = B_{F0}(L_{WL} + 4.1 \frac{u_r}{\omega_{roll} L_{WL}}) p_r + B_L p_r u_r, \tag{3.18}$$

where B_{F0} is the linear friction at $u = 0$, B_L is the lift damping, L_{WL} is the water line length and ω_{roll} is the frequency of the motion in roll. Eddy damping, caused by flow separation at the bottom of the ship hull due to roll motion, has not been included as it is negligible when the Froude number is larger than 0.2, which for a small to medium sized sailboat can safely be neglected.

3.1.6 Sail, Keel and Rudder

Foils

The sail, keel and rudder are all modeled using foils. The lift and drag of a foil are calculated using the following formulas:

$$F_L = \frac{1}{2} pAC_L(\alpha)V^2 \tag{3.19}$$

$$F_D = \frac{1}{2} pAC_D(\alpha)V^2, \tag{3.20}$$

where p is the density of the fluid (in this context, air or water), A is the area of the foil, V is the speed of the fluid over the foil and $C_L(\alpha)$ and $C_D(\alpha)$ are the lift and drag coefficients which are functions of the angle of attack, α. The velocity of the fluid passing by a foil on the boat is equal to[3]

$$V_f = J_{2D}^T(\psi, \phi)V_{w/c} - \nu_{uv} - \begin{bmatrix} -ry'_f \\ rx'_f - pz'_f \end{bmatrix} - \begin{bmatrix} \dot{x}'_f \\ \dot{y}'_f \end{bmatrix}, \tag{3.21}$$

where x_f and z_f is the x' and z' position of the foil in relation to CO. $V_{w/c}$ is the wind/current vector.

Sail

The speed of the wind that is passing by the sail is calculated by (3.21):

$$V_{ws} = J_{2D}^T(\psi, \phi)V_w(-h_1 cos(\phi)) - \nu_{uv} - \begin{bmatrix} rl_2 sin(\lambda) \\ -r(l_1 + l_2 cos(\lambda)) + ph_1 \end{bmatrix} - \begin{bmatrix} \dot{\lambda} l_2 sin(\lambda) \\ -\dot{\lambda} l_2 cos(\lambda) \end{bmatrix}, \tag{3.22}$$

where $V_w(-h_1 cos(\phi))$ is the wind speed at $h_1 cos(\phi)$ meters above the sea level, which is the the center of effort of the sail. The angle of the wind can then be calculated as[4]

$$\beta_{ws} = arctan2(V_{ws_v}, V_{ws_u}), \tag{3.23}$$

and the angle of attack as

$$\alpha_s = \beta_{ws} - \lambda + \pi \cdot \tag{3.24}$$

We can then calculate S_L (lift of sail) and S_D (drag of sail) using formula (3.19), (3.20) and (3.24). Using trigonometry and the definitions of figure 3.1, S can be formulated by

[3] When subscripts u, v, x', y' (etc) are added to a variable it means that the u or v component(s) of that vector is being used. Example: $\nu_{uv} = [u \ v]^T$.

[4] $arctan2(y, x) \in [-\pi, \pi]$ is the four-quadrant inverse tangent.

$$S = \begin{bmatrix} -S_L sin(\beta_{ws}) + S_D cos(\beta_{ws}) \\ S_L cos(\beta_{ws}) + S_D sin(\beta_{ws}) \\ h_1(S_L cos(\beta_{ws}) + S_D sin(\beta_{ws})) \\ S_{x'} l_2 sin(\lambda) - S_{y'}(l_1 + l_2 cos(\lambda)) \end{bmatrix}, \tag{3.25}$$

where $S_{x'}$ and $S_{y'}$ are the forces from S in x' and y' direction respectively.

Keel

For the keel we follow a similar approach. As with the wind, we first have to find the relative speed of the water to the ship:

$$V_{ck} = -\nu_{r_{uv}} - \begin{bmatrix} 0 \\ -rl_3 - ph_2 \end{bmatrix} \cdot \tag{3.26}$$

The angle the water passing by the keel can be calculated as

$$\beta_{ck} = arctan2(V_{ck_v}, V_{ck_u}) \cdot \tag{3.27}$$

Then we find the angle of attack using

$$\alpha_k = -\beta_{ck} + \pi \cdot \tag{3.28}$$

Notice that the angle of attack is negative of β_{cr}. This is due to the definition of direction of forces (see figure 3.1).

K_L and K_D can now be calculated using (3.19), (3.20) and (3.28). This gives K as:

$$K = \begin{bmatrix} K_L sin(\beta_{ck}) + K_D cos(\beta_{ck}) \\ -K_L cos(\beta_{ck}) + K_D sin(\beta_{ck}) \\ h_2(K_L cos(\beta_{ck}) - K_D sin(\beta_{ck})) \\ l_3(K_L cos(\beta_{ck}) - K_D sin(\beta_{ck})) \end{bmatrix} \cdot \tag{3.29}$$

Rudder

The equation of the water speed relative to the rudder is

$$V_{cr} = -\nu_{r_{uv}} - \begin{bmatrix} 0 \\ -rl_4 - ph_3 \end{bmatrix} \cdot \tag{3.30}$$

The angle of the relative speed of the water passing by can be calculated as:

$$\beta_{cr} = arctan2(V_{cr_v}, V_{cr_u}) \cdot \tag{3.31}$$

The angle of attack of the rudder can be found by:

$$\alpha_r = -\beta_{cr} + \sigma + \pi \cdot \tag{3.32}$$

R is found in the same manner as we found S and K:

$$R = \begin{bmatrix} R_L sin(\beta_{cr}) + R_D cos(\beta_{cr}) \\ -R_L cos(\beta_{cr}) + R_D sin(\beta_{cr}) \\ h_3(R_L cos(\beta_{cr}) - R_D sin(\beta_{cr})) \\ l_4(R_L cos(\beta_{cr}) - R_D sin(\beta_{cr})) \end{bmatrix} . \tag{3.33}$$

3.1.7 Actuators

Rudder

The rudder can not move or accelerate infinitely fast and the maximum angle is restricted due to physical limitations. In the model this is accomplished by setting the maximum allowed $|\sigma| \leq \sigma_{sat}$. A first order low-pass filter with a time constant T_r is also applied to the input signal, effectively causing a delay when moving the rudder. The transfer function of a first order low-pass filter is

$$F(s) = \frac{\omega}{s + \omega}, \tag{3.34}$$

where

$$\omega = \frac{1}{T} \tag{3.35}$$

and T is the time constant of the filter.

Sail

The sail actuator has to be modeled more accurately as it is such an important part of the system. It is assumed that the rope connected to the boom has no mass and that the boom dynamics happens in a two-dimensional environment, and that the body frame is inertial. The boom dynamics are a lot faster than the dynamics of the body-frame, which makes the body frame behave as it is inertial in this context. The position of the sail is controlled by the wind and a rope controlling maximum λ. When the rope is being stretched it is modeled by a spring and a dampening component. The boom will be modeled like a beam with a center of gravity in the middle, with friction between the mast and the boom. Using all of the approximations the system dynamics can be described as in Figure 3.4.

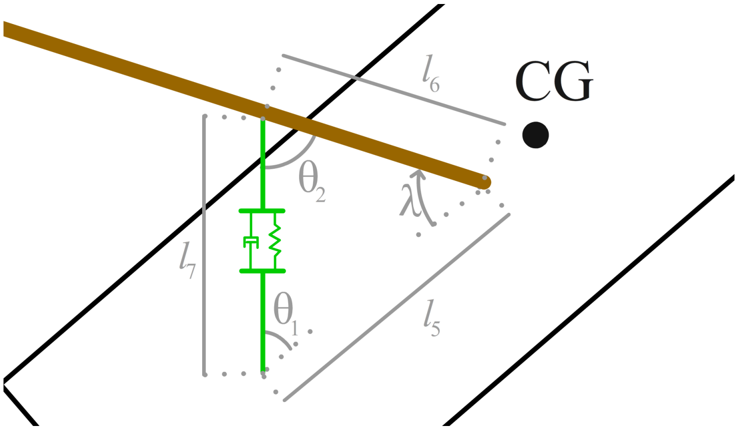


Figure 3.3: Boom dynamics, definitions of important lengths and angles

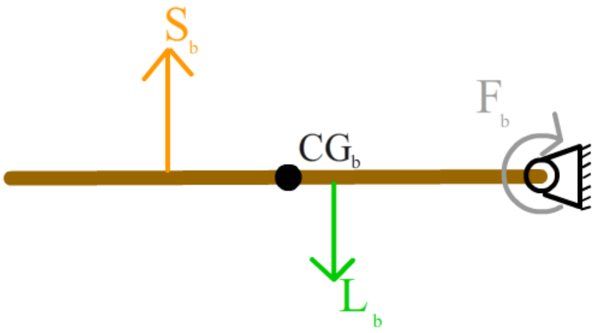


Figure 3.4: Boom dynamics, definitions of important lengths and angles cont.

The rotational dynamics of the boom can then be expressed by the following equation:

$$\ddot{\lambda} I_b = l_2 S_b - l_6 L_b + F_b, \tag{3.36}$$

where I_b is the rotational inertia of the boom, S_b is the force generated by the sail perpendicular to the boom, L_b is the force created perpendicular to the boom by the rope, and F_b is the torque created by friction between the boom and the mast. S_b can be calculated by knowing $S_{x'}$, $S_{y'}$ and λ:

$$S_b = l_2(S_{x'} sin(\lambda) - S_{y'} cos(\lambda)) \cdot \tag{3.37}$$

F_b is modeled by a linear friction component, where F_r is the friction coefficient:

$$F_b = -\dot{\lambda} F_r \cdot \tag{3.38}$$

L_b can be calculated by:

$$L_b = -l_6 Q sin(\theta_2), \tag{3.39}$$

where $Q \in [-\infty, 0]$ is the force created by the stretching of the rope. The length of the rope while not stretched will be referred to as l_{7_0}, while the true length will be referred to as l_7. The true length can be found by

$$l_7 = \sqrt{dX^2 + dY^2}, \tag{3.40}$$

where

$$dX = l_5 - l_6 cos(\lambda), \tag{3.41}$$

$$dY = -l_6 sin(\lambda) \cdot \tag{3.42}$$

θ_2 can be calculated using

$$l_5^2 = l_6^2 + l_7^2 - 2 l_6 l_7 cos(\theta_2), \tag{3.43}$$

from which it follows

$$\theta_2 = acos(\frac{l_6^2 + l_7^2 - l_5^2}{2 l_6 l_7}) \cdot \tag{3.44}$$

Δ_L is the length of which the rope is stretched, defined by

$$\Delta_L = l_7 - l_{7_0} \cdot \tag{3.45}$$

$\dot{\Delta}_L$ can then be calculated as as

$$\begin{aligned} \dot{\Delta}_L &= \dot{l_7} - \dot{l_{7_0}} \\ &= \dot{\sqrt{dX^2 + dY^2}} - \dot{l_{7_0}} \\ &= \frac{dX \dot{dX} + dY \dot{dY}}{\sqrt{dX^2 + dY^2}} - \dot{l_{7_0}}, \end{aligned} \tag{3.46}$$

where

$$\dot{dX} = l_6 \dot{\lambda} sin(\lambda) \tag{3.47}$$

$$\dot{dY} = -l_6 \dot{\lambda} cos(\lambda) \cdot \tag{3.48}$$

Q can then be expressed as

$$Q = -\Delta_L Q_k - \dot{\Delta}_L Q_b, \tag{3.49}$$

where Q_k is the spring coefficient of the rope and Q_b is the dampening coefficient.

The actuator controlling the length of the rope also have physical limitations, such as maximal torque, friction and inertia. However, to make implementation simpler, this actuator is modeled as a first order low-pass filter with a time constant of T_b. A limitation on maximum angle of the sail is also set as $|\lambda| \leq \lambda_{sat}|$.

3.1.8 Wind

The wind is simulated by the sum of a mean component and a gust component. The mean wind speed at 10 meter above the sea level will be referred to as U_{10m}. The guest is estimated using the following wave spectra (NORSOK standard [2007]):

$$S(f) = 320\frac{(\frac{U_{10m}}{10})^2(\frac{-z}{10})^{0.45}}{(1+x^n)^{\frac{5}{3n}}} \tag{3.50}$$

$$x = 172f\frac{-z}{10}^{\frac{2}{3}}(\frac{U_{10m}}{10})^{\frac{-3}{4}} \tag{3.51}$$

$$\delta = \sqrt[2]{2S(f)\Delta f}cos(2\pi ft + \Upsilon), \tag{3.52}$$

where f is the frequency of the gust, δ is the amplitude of the gust, t is the time, $z = -10$ is the height above sea level, n is a constant equal to 0.468 and Υ is an evenly distributed phase for each f. Frequencies between 0 and 0.4 Hz will be simulated as this covers most of the energy in the spectrum. Υ is kept equal between simulations to keep randomness out of the final conclusion. We also have to correct for the wind speed being lower depending on how high we are above the sea level

$$\begin{aligned} U(z) &= U_{10}\frac{5}{2}\sqrt{k}\log\frac{-z}{z_0} \\ z_0 &= 10e^{-\frac{2}{5\sqrt{k}}}, \end{aligned} \tag{3.53}$$

where U_{10} is the sum of the mean and gust wind speed at 10 meters above sea level, $U(z)$ is the wind speed at z meters and $k = 0.0026$. $V_w(z)$ is then given by

$$V_w(z) = U(z)\begin{bmatrix} cos(\beta_w) \\ sin(\beta_w) \end{bmatrix} \tag{3.54}$$

In addition to the gust, the wind direction will fluctuate, which is modeled by

$$\begin{aligned} \beta_w &= \beta_{w0} + \beta_{w_{flux}} \\ \dot{\beta}_{w_{flux}} &= w, \end{aligned} \tag{3.55}$$

where β_{w0} is the average wind direction and w is white noise. $\beta_{w_{flux}}$ is saturated at 5 degrees.

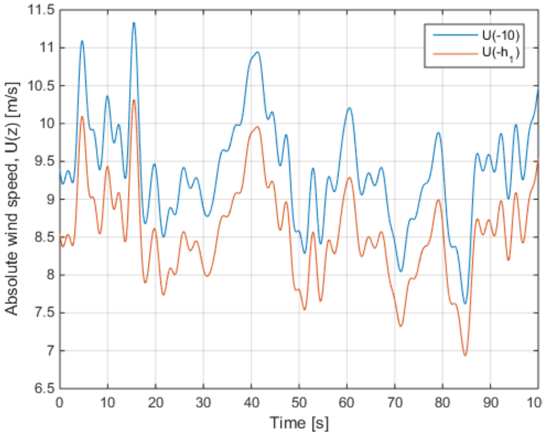


Figure 3.5: Time evolution of $U(z)$

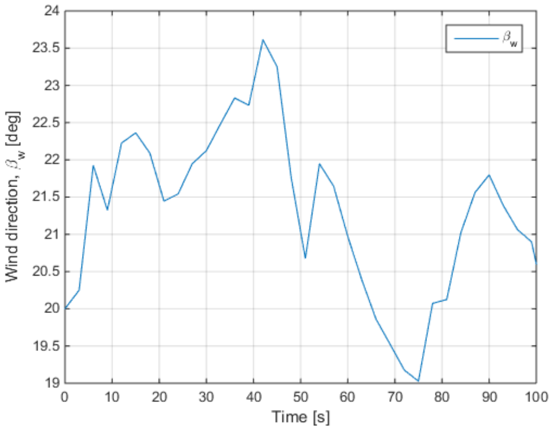


Figure 3.6: Time evolution of β_w

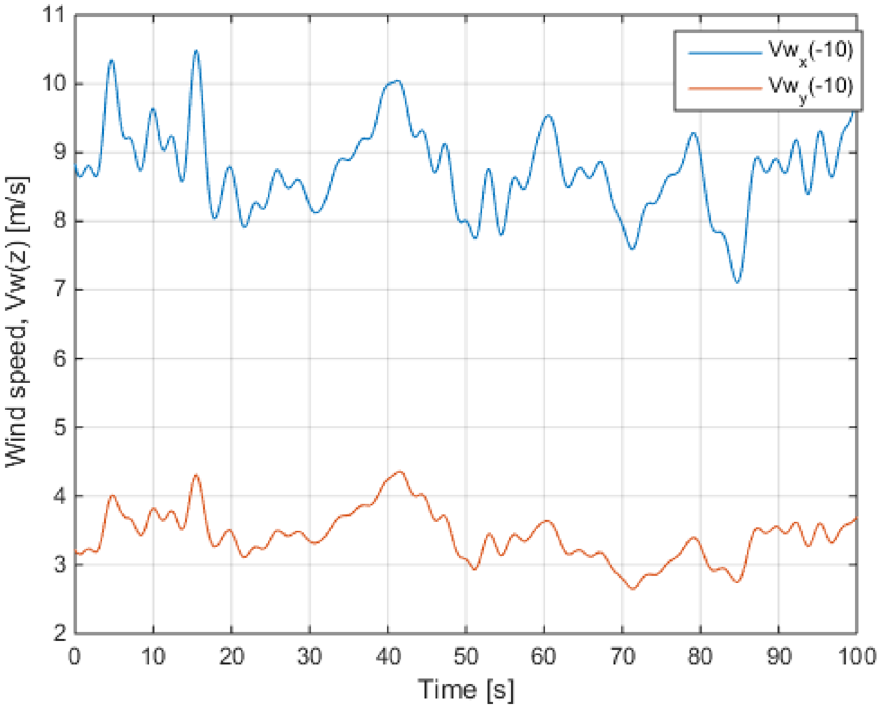


Figure 3.7: Wind simulation

3.2 Estimating Simulation Parameters

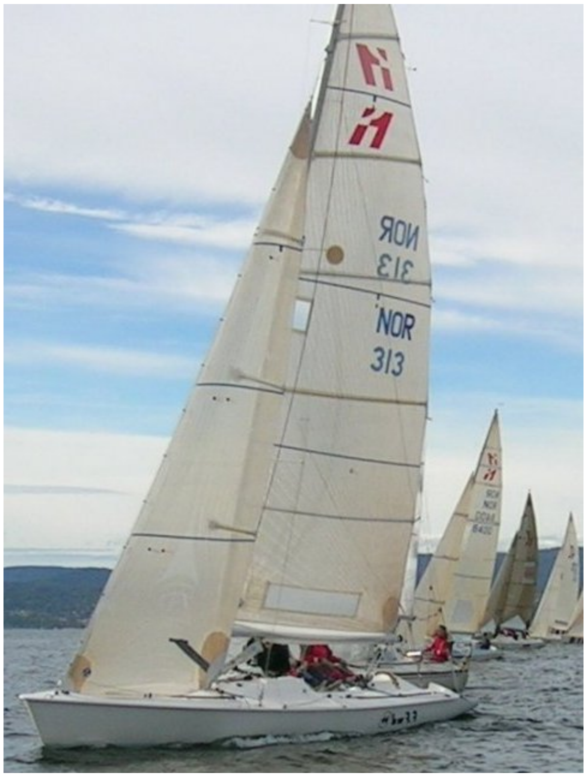


Figure 3.8: Picture of boat to be modeled

The following calculations are based on the data in Appedix B. The data includes some basic information and a couple of drawing of the ship. Using this we can estimate the variables we need for the model. In all of the following calculations it is assumed that the mass of the ship is evenly distributed by volume, and that strip theory is valid (sleek body).

Accurate values are very hard and time consuming to calculate. Most of the following values are rough estimates. The object of this thesis is first and foremost to uncover the dynamics of a generalized sailboat such that new control approaches can be tested and discovered, not to model this particular sailboat as accurately as possible.

3.2.1 Shape of sailboat

Before we can do any calculations on the sailboat we need to find a way to describe the hull by equations. By looking at the top down view of the hull (Appendix B) and recording the beam length at certain points along the hull, one can make a good approximation by using linear interpolation between each data point. The same procedure is used to make an approximation of the draft by using the side view of the boat.

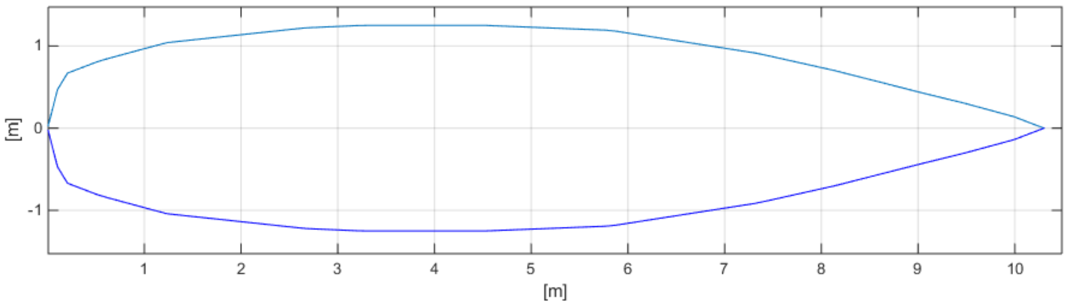


Figure 3.9: Plot of beam

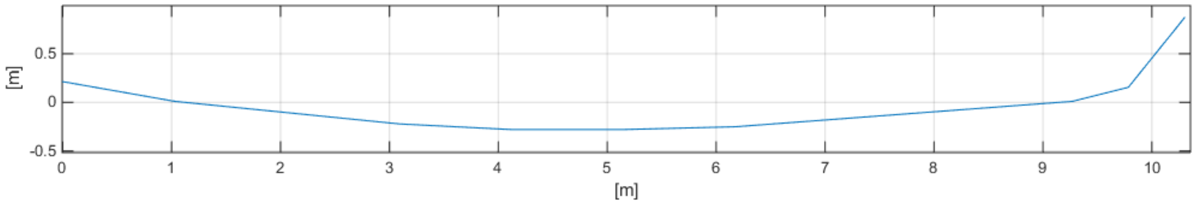


Figure 3.10: Plot of draft

In Figure 3.9 and 3.10 the stern of the hull is at $x = 0$, and in Figure 3.10 the water line is at $y = 0$ (the y-axis in Figure 3.10 is equal to the negative z-axis in the n-frame, which is done for illustrative purposes). As an approximation of the beam length at a given height, one can assume that the shape of the hull can be described by the following equations:

$$\begin{aligned} h &= d(x')\ sin(a), \\ b &= B(x')\sqrt{cos(a)}, \end{aligned} \tag{3.56}$$

where $d(x')$ is the "draft" (measured from the top of the hull, not waterline), $B(x')$ is half of the beam length, and h and b is explained in Figure 3.11.

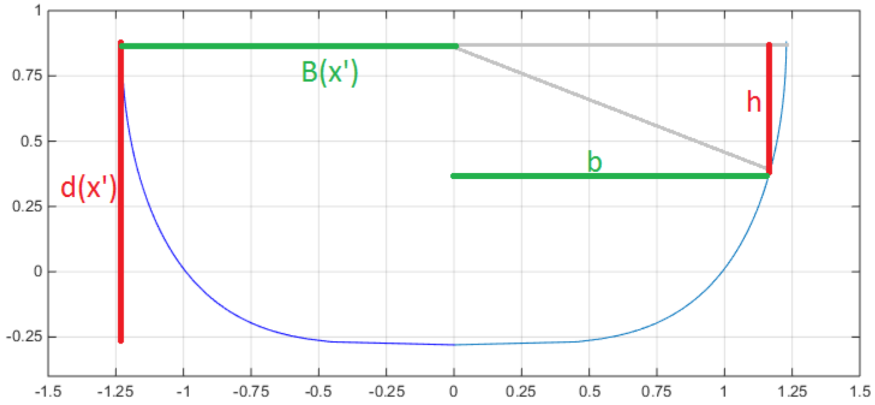


Figure 3.11: Plot of cross section

We can then solve (3.56) on b for a given h and x' as

$$b = B(x')\sqrt{cos(asin\Big(\frac{h}{d(x')}\Big))} = B(x')\sqrt[4]{1 - \Big(\frac{h}{d(x')}\Big)^2} \tag{3.57}$$

3.2.2 Mass

Rigid body

m is equal to the displacement of the sailboat, which is 1600kg. Of the 1600kg, 700kg is ballast. The following assumptions are used about the mass distribution of the different parts:

- boom weight, 10 $[kg]$
- mast weight, 25 $[kg]$
- keel weight, 150 $[kg]$
- ballast is located in the bulb of the keel
- hull weight is equal to 865 (displacement minus boom, mast and ballast)

CG_x and CG_z of the hull can be approximated by

$$hull\ CG_x = \frac{\int_V xdV}{V} \tag{3.58}$$

$$hull\ CG_z = \frac{\int_V zdV}{V}, \tag{3.59}$$

where V is the volume of the hull. This calculation is done numerically, using the shape of the hull as described in Figure 3.9, 3.10 and 3.11.

From the picture of the sailboat (Appendix B) one can easily require all the necessary heights and positions of all the parts to make an estimate of the center of gravity. In the table below, all lengths are measured from the stern, and height measured from the water line.

Variable	**value**
mast CG_z	-5 $[m]$
ballast CG_z	1.6 $[m]$
keel CG_z	0.95 $[m]$
boom CG_z	-5 $[m]$

Variable	**value**
mast CG_x	6.27 $[m]$
bulb CG_x	5.1 $[m]$
keel CG_x	5.1 $[m]$

Table 3.5: center of gravity of mast, keel and boom

Using all of the above information, CG_x and CG_z of the boat can be found by

$$CG_x = \frac{\sum_{i=0}^{n} \text{weight of element i} \times CG_x \text{ of element i}}{\text{total weight}} [m] \tag{3.60}$$

and

$$CG_z = \frac{\sum_{i=0}^{n} \text{weight of element i} \times CG_z \text{ of element i}}{\text{total weight}} \cdot \tag{3.61}$$

I_{xx} and I_{zz} of the hull are calculated by

$$\frac{I_{xx}}{\text{hull weight}} = \frac{\int_V x'^2 dV}{V} \tag{3.62}$$

$$\frac{I_{zz}}{\text{hull weight}} = \frac{\int_V (z - CG_z)^2 dV}{V}, \tag{3.63}$$

I_{xx} and I_{zz} can then be found for the sailboat, assuming roll motion about the water plane area and yaw motion about CG_x [5]:

$$I_{xx} = \sum_{i=0}^{n} \text{weight of element i} \times x'^2 + \text{local rotational inertia about x-axis} \tag{3.64}$$

and

$$I_{zz} = \sum_{i=0}^{n} \text{weight of element i} \times z_i^2 + \text{local rotational inertia about z-axis} \cdot \tag{3.65}$$

[5] The local rotational inertia is assumed small (≈ 0) for the mast and keel.

Added mass

Calculation of added mass is complicated, especially for three-dimensional objects near (or at) the free surface. However, strip theory (Faltinsen [1990], pp.41-58), which can be used for slender bodies, is a simplified approach for roughly estimating the three-dimensional added mass coefficients by looking at two-dimensional strips. The two-dimensional added mass coefficient for a cylinder (at a frequency of 0) in water, half of it submerged, can be expressed as

$$A_c(R) = \frac{1}{2} p_w \pi R^2, \tag{3.66}$$

where R is the radius of the cylinder. The added mass coefficient for a flat plate, which will be used to approximate the keel and rudder, is equal to

$$A_p(w) = p_w \pi (\frac{w}{2})^2, \tag{3.67}$$

where w is the height of the flat plate. $Y_{\dot{v}}(0)$, $K_{\dot{p}}(0)$, $N_{\dot{r}}(0)$ and $Y_{\dot{r}}(0)$ can then be calculated by using the following equations:

$$N_{\dot{r}}(0) = -A_p(k_h) k_w h_2^2, \tag{3.68}$$

$$Y_{\dot{v}}(0) = -\int_{LWL} A_c(T(x'))dx' - A_p(k_h)k_w - A_p(r_h)r_w, \tag{3.69}$$

$$N_{\dot{r}}(0) = -\int_{LWL} A_c(T(x'))x'^2 dx' - A_p(k_h)k_w(-l_3)^2 - A_p(r_h)r_w(-l_4)^2, \tag{3.70}$$

$$Y_{\dot{r}}(0) = -\int_{LWL} A_c(T(x'))x'dx' + A_p(k_h)k_w l_3 + A_p(r_h)r_w l_4, \tag{3.71}$$

where $T(x')$ is the draft at x', k_h is the height of the keel, k_w is the width of the keel, r_h is the height of the rudder and r_w is the width of the rudder.

Strip theory can not be used to calculate the added mass in surge, $X_{\dot{u}}(\infty)$. Instead, a typical value equal to 10% of the displacement of the ship is chosen. The added mass in surge is low compered to the mass of the ship and does not contribute as much to the system dynamics as $Y_{\dot{v}}$, $K_{\dot{p}}$ and $N_{\dot{r}}$.

3.2.3 Restoration forces

Equation (4.32) in (Fossen [2011],p. 65) is used to calculate GM_t:

$$\begin{aligned} GM_t &= BM_t - BG \\ BM_t &= \frac{I_T}{\Delta} = \frac{\int_{A_{wp}} y'^2 dA}{\Delta} \\ BG &= CB - CG_z, \end{aligned} \tag{3.72}$$

where CG_z is the center of gravity, CB is the center of bouncy, I_L is the moment of area about the water plane and A_{wp} is the water plane area. CB is assumed to be equal to a third of the maximum draft for easy calculation.

3.2.4 Dampening

Linear dampening

Fossen [2011] proposes the following formulas to approximate the linear dampening terms:

$$d_{l_{11}} = \frac{m_{11} - X_{\dot{u}}(0)}{T_{surge}} \tag{3.73}$$

$$d_{l_{22}} = \frac{m_{22} - Y_{\dot{v}}(0)}{T_{sway}} \tag{3.74}$$

$$d_{l_{66}} = \frac{I_{zz} - N_{\dot{r}}(0)}{T_{yaw}}, \tag{3.75}$$

where typical values for T_{surge}, T_{sway} and T_{yaw} are 100-250s. The median value of 175s is chosen for T_{surge}, T_{sway} and T_{yaw}, but further effort could be done to better estimate this value. Accurate modeling of linear drag is mostly important for sway and yaw motion as the quadratic term dominates in surge, thus $X_{\dot{u}}(\infty)$ is used in stead of $X_{\dot{u}}(0)$ even though this approximation is not very accurate.

Non-linear Dampening

$D_{q_X}(\nu_r)$, equation (3.14), is dependent on the wetted surface area, S_H, and the form factor coefficient, k. S_H is estimated by

$$S_H = L_{WL}(1.7T_{max} + C_B B_{WL}), \tag{3.76}$$

$$C_B = \frac{\Delta}{L_{WL} B_{WL} d}, \tag{3.77}$$

where T_{max} is the maximum draft of the hull, C_B is the block coefficient, Δ is the displacement of the boat and B_{WL} is the beam width at the water line.

$D_{q_Y}(\nu_r)$ and $D_{q_N}(\nu_r)$, respectively equation (3.16) and (3.17), is dependent on the draft, $T(x')$, and the two-dimensional drag coefficient $C_D(x')$. $C_D(x')$ is estimated by using Hoerner's curve (Hoerner [1965]), see appendix A. To increase simulation speed, $C_i = C_D(x_i')T(x_i')dx'$ is pre-calculated at ten evenly distributed points along the ship, and equation (3.16) and (3.17) are modified to be computed numerically at thees ten points:

$$D_{q_Y}(\nu_r) = \frac{1}{2}p_w \sum_{i=0}^{9} C_i |v_r + x_i' r|(v_r + x_i' r), \tag{3.78}$$

$$D_{q_N}(\nu_r) = \frac{1}{2}p_w \sum_{i=0}^{9} C_i x_i' |v_r + x_i' r|(v_r + x_i' r)\cdot \tag{3.79}$$

The roll dampening, $D_{q_K}(\nu_r)$ equation (3.18), is dependent on B_{F0}, B_L and ω_{roll}. ω_{roll} is simplified by using the natural frequency in roll, ω_{roll_n}, which can be estimated by

$$\omega_{roll_n} = \sqrt{\frac{GM_t\Delta}{I_\phi - K_{\dot{p}}}}\cdot \tag{3.80}$$

B_{F0} is the linear friction coefficient at zero speed and B_L is the lift dampening coefficient. To see how thees coefficients can be calculated in more detail, see Himeno [1981], however, the main equations are presented below.

$$\begin{aligned} B_{F0} &= 0.787 p_w S_H r_s^2 \sqrt{\omega_{roll_n}\nu_w}, \\ r_s &= \frac{B_{WL} + T_{max}}{2}, \end{aligned} \tag{3.81}$$

where ν_w is the kinematic viscosity of water.

$$\begin{aligned} l_0 &= 0.3T_{max}, \\ l_R &= 0.5T_{max}, \\ k_N &= 2\pi\frac{T_{max}}{L_{WL}} + k(4.1\frac{beam_W L}{L_{WL}} - 0.045), \\ B_L &= \frac{1}{2}p_w L_{WL} T_{max} k_N l_0 l_R (1 - 1.4\frac{CG_z}{l_R} + 0.7\frac{CG_z}{\frac{l_0}{l_R}}), \end{aligned} \tag{3.82}$$

where k is dependent on how circular the cross section of the hull is, and k is equal to zero in this case. See original source material for more information on how k is calculated.

3.2.5 Sail keel and rudder

Lift and drag coefficients for the sail were found in C.A.Marchaj [2000], p. 587. The data of the lift and drag coefficients of the sail are gathered using real data, though it is probably a big simplification of the real world. It's reasonable to believe that the lift and drag of the sail would be dependent on states such as wind speed, twist, the vertical camber and the draft, which are not represented in the model. The drag and lift coefficients for the rudder are based on NACA0015, while the keel is based on NACA0009 (Sheldahl and Klimas [1981]). See appendix A for plots of the lift and drag coefficients.

The lift of a 2-dimensional flat plate is approximately equal to

$$C_{L_{2D}}(\alpha) = 2\pi\alpha \cdot \tag{3.83}$$

According to Wagner [1948], this formula can be modified for a finitely long plate to be

$$C_L(\alpha) = \frac{2\pi\alpha}{1+\frac{2}{Asp}} = \frac{C_{L_{2D}}(\alpha)}{1+\frac{2}{Asp}} \cdot \tag{3.84}$$

where Asp is the aspect ratio of the foil, defined as

$$Asp = \frac{span}{chord} \cdot \tag{3.85}$$

For a foil that is similar in shape to a flat plate, such as NACA0015 and NACA0009, one can assume that (3.84) still holds true. Assuming this one can recalculate the lift coefficients for an arbitrary aspect ratio:

$$C_L(\alpha) = \frac{C_{l0}(\alpha)(1+\frac{2}{Asp_0})}{(1+\frac{2}{Asp})}, \tag{3.86}$$

where Asp_o is the original aspect ratio of the foil used when calculating the lift coefficient, C_{l0} is the original lift coefficient, and Asp is the aspect ratio of the foil we want to scale the results to. The drag coefficients for the rudder and keel only account for the induced drag $C_{Di}(\alpha)$ and do not take into account the viscus drag. The viscus drag are added to the drag coefficients:

$$C_D(\alpha) = C_{Di}(\alpha) + 2(1+2\frac{t_{max}}{c})C_F(R_n), \tag{3.87}$$

where t_{max} is the thickness of the foil and c is the chord length.

3.2.6 Boom

No paper before have tried to model the boom as accurately, and thus finding good values were difficult. The parameters F_r, Q_k and Q_b were thus instead manually tuned to give a response that looked real. This was very difficult to do for F_r as it does not contribute as much to the dynamic as the other parameters, and because the sail itself contributes most of the dampening. Maybe in future works one can try to determine the value of F_r, but currently it is set equal to zero.

3.2.7 Other

The lengths defined in Figure 3.1 and 3.2, and the area and aspect ratios of the keel and rudder, where found by looking at the drawings of the sailboat (appendix B).

3.3 Results

3.3.1 Boom

The first simulation will focus on the dynamics of the boom. $\dot{\nu} = \nu = 0$ to ensure that the boat is standing still, and wind gusts and the wind direction changing are disabled. $\beta_{ws} = 20deg$, wind speed of $5\frac{m}{s}$, $l_{70} = 2m$ and the initial value of λ is zero.

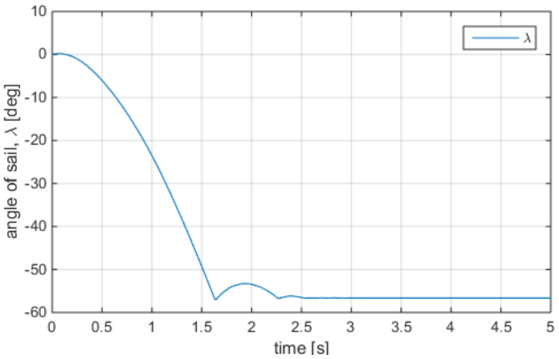


Figure 3.12: Boom simulation, λ

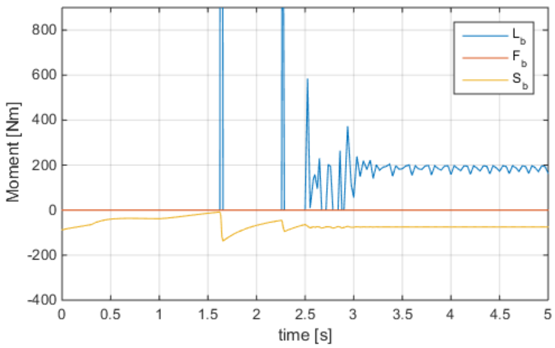


Figure 3.13: Boom simulation, moments

Figure 3.12 shows the time evolution of λ. The sail accelerates quickly in the beginning and stops just as quickly when the rope tightens. The spring effect of the rope make the boom bounce a couple of times before the movement stops completely. In Figure 3.13 one can see that it is S_b that creates the big acceleration at the beginning. This acceleration then gets smaller as the sail begins to move in the same direction as the wind, effectively reducing the speed as experienced by the sail. When the rope tightens, large spikes in L_b is observed. The two first spikes goes beyond the graph, and reaches a maximum of roughly $2000Nm$. Afterwards L_b becomes zero, and S_b increases in size again due to the sail now moving against the wind. This pattern is then repeated until S_b and L_b reaches steady values.

3.3.2 Forces while running

Running describes the maneuver of sailing down-wind. In this simulation the wind speed is $5\frac{m}{s}$, and the controller developed in chapter 4 is used to keep the boat heading the correct course angle.

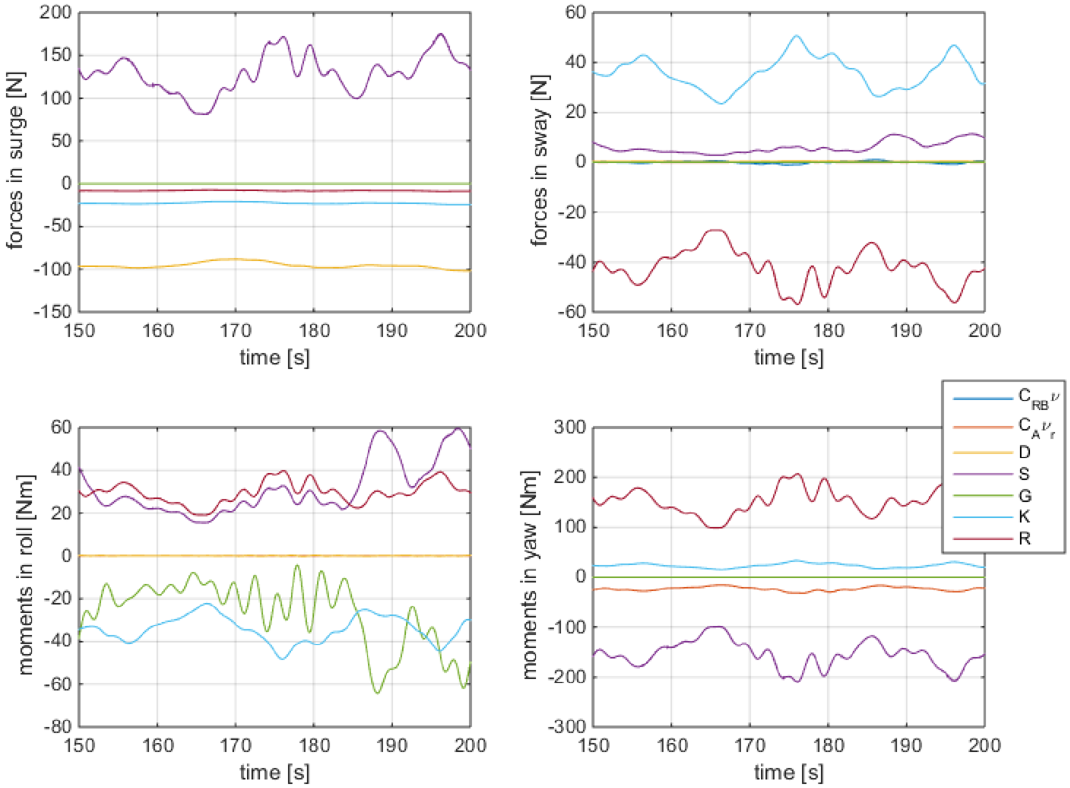


Figure 3.14: Running simulation, forces.

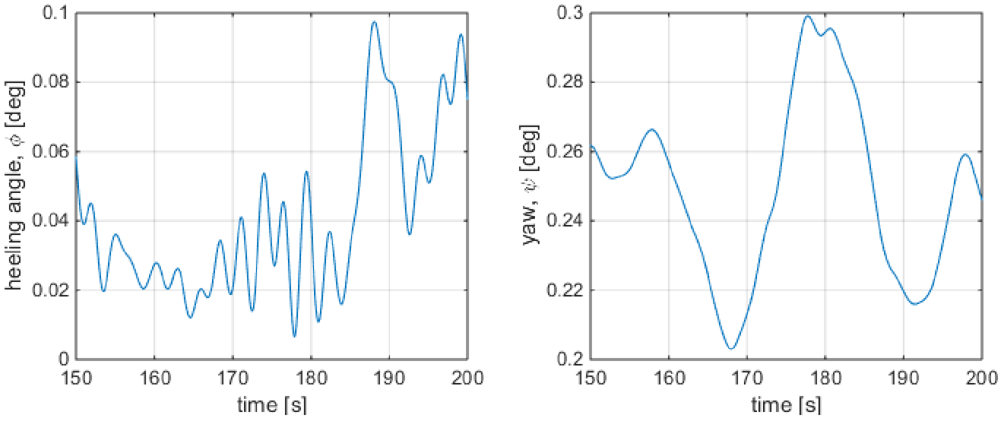


Figure 3.15: Running simulation, heeling angle and yaw.

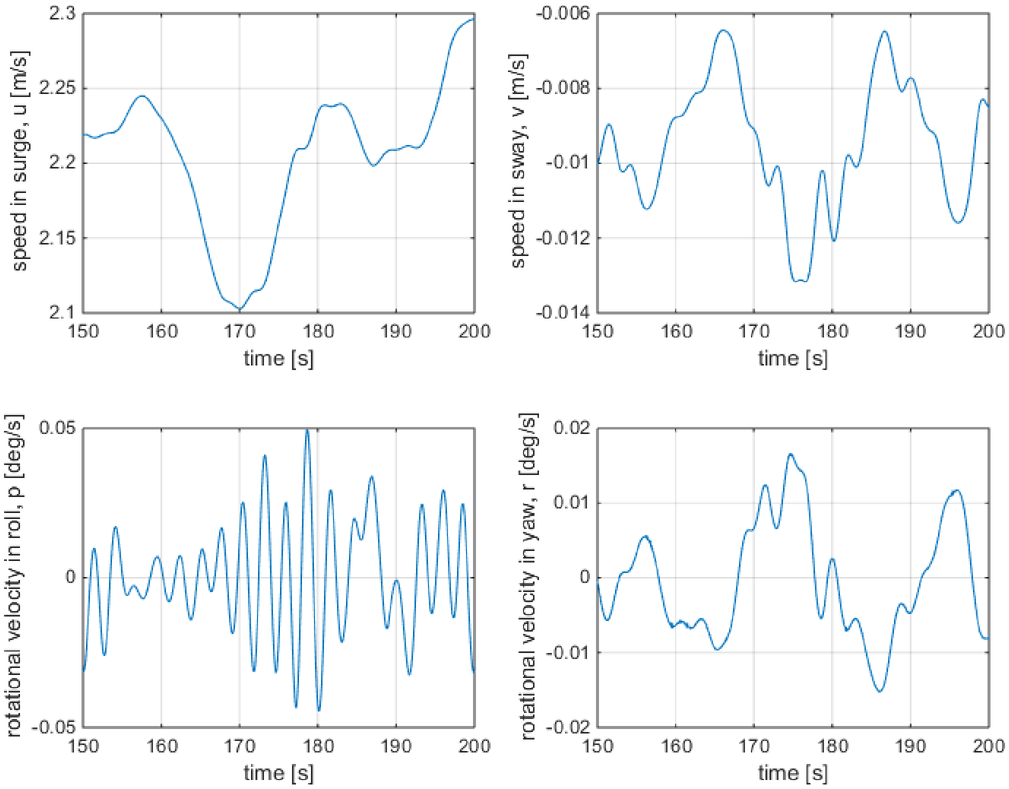


Figure 3.16: Running simulation, ν.

In Figure 3.14 shows the forces that acts on the boat while running. In surge one can see that the sail and the drag are dominating, addition to the drag caused by the keel. In sway, the rudder and keel causes most of the forces, but also the sail is contributing. If the keel is producing forces in sway one can assume that the ship has a small drift angle, which indeed can be found by looking at the ratio between v and u in Figure 3.16.

The forces in roll are small compared to the inertia and added mass on this axis of motion, which is not to surprising while running. Looking at Figure 3.15 one finds that the heeling angle is very small as well. In yaw the sail and rudder are dominating, but small contributions from the keel and from the coriolis added mass is seen as well.

3.3.3 Forces: Beam reaching

Beam reaching describes the maneuver of sailing perpendicular to the wind. In this simulation the wind speed is $5\frac{m}{s}$, and the controller developed in chapter 4 is used to keep the boat heading the correct course angle.

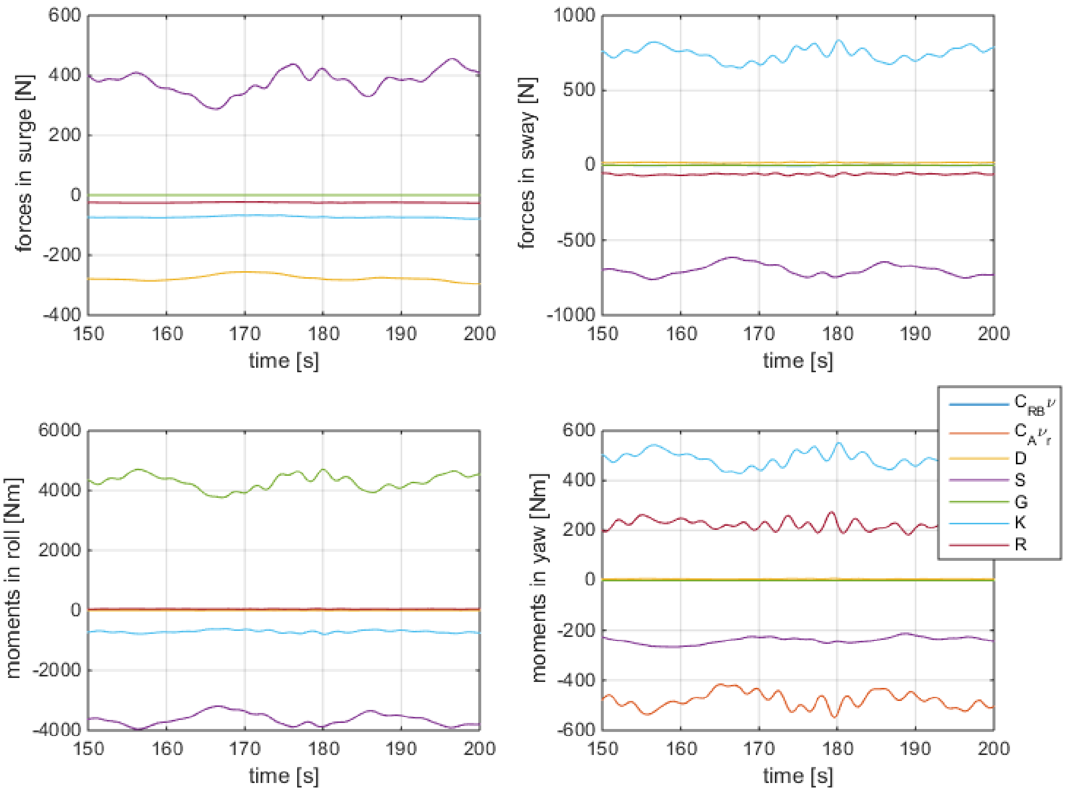


Figure 3.17: Beam reaching simulation, forces.

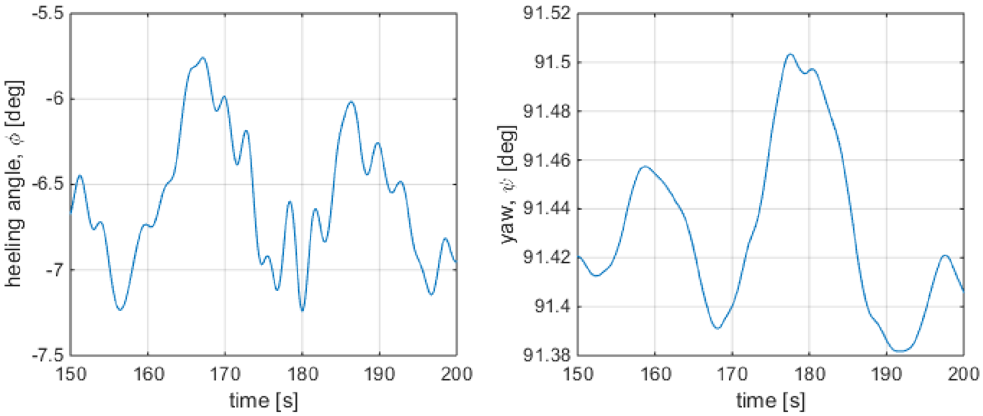


Figure 3.18: Beam reaching simulation, heeling angle and yaw.

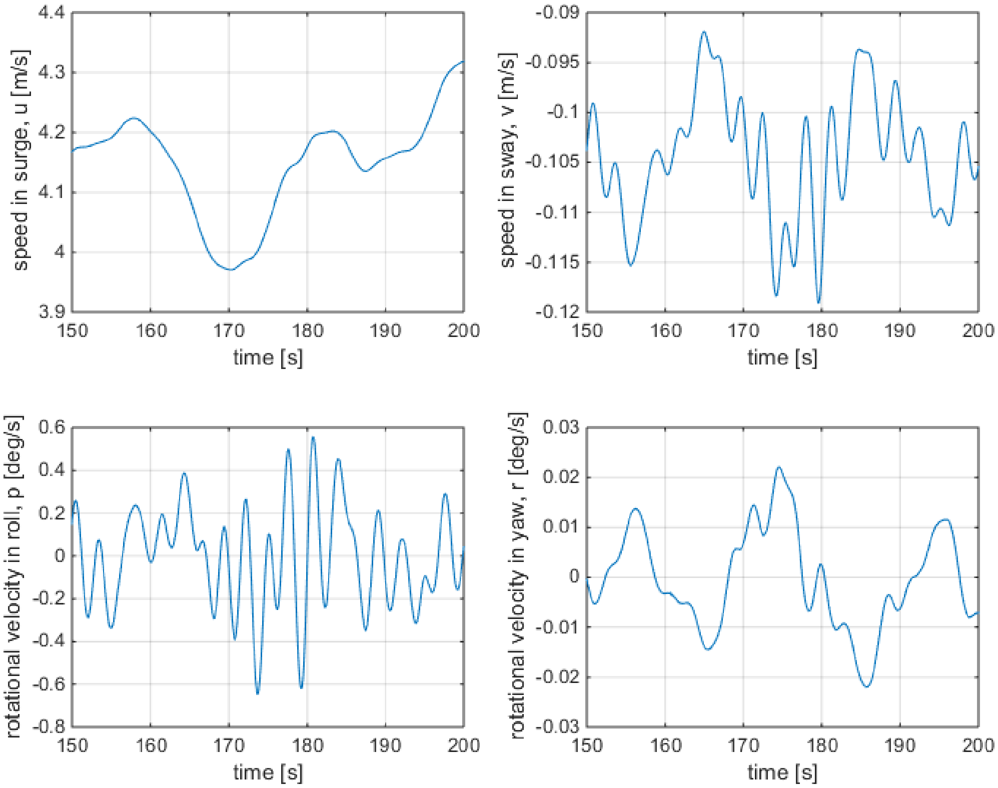


Figure 3.19: Beam reaching simulation, ν.

The first plot in Figure 3.17 shows that the same forces are dominating in surge while running as when beam reaching. The same can not be said for the forces in sway, where the sail and keel are now dominating. Looking at Figure 3.19 one finds that the speed in sway has greatly increased, and it follows that the drift angle has increased as well. The increase in side force caused by the sail is obliviously caused by the wind hitting the boat from the side, compared to the stern as when running.

The increase of forces in sway also causes an increase in the moments in roll. The restoring moment and sail are dominating, though the keel is also a key force. The increase in the restoring moment is due to the high heeling angle, which can be seen in Figure 3.18. The dampening moment in roll is very small, almost so that it can not be seen. The hull causes little to no dampening in roll, and does not contribute much to the dynamic. In yaw, the coriolis added mass, sail, keel and rudder are contributing a lot of moments. Compared to running, the forces and moments are larger on all of the axis, indicating that beam reaching introduces a lot more stress on the boat.

4 The ideal controller

The purpose of this chapter is to describe an ideal controller which assumes perfect knowledge about the system and the states of the system at any time. This is partly to demonstrate what theoretically can be achieved with a control system on a sailboat, but first and foremost to have something to compare against when a more practical controller is developed later on in the thesis. The goal of the controller is to be able to follow a course at maximum speed, but at the same time limiting dangerously large heeling motions.

The sailboat has two main actuators: The sail and the rudder. The system has four degrees of freedom (4DOF) but only two actuators, meaning that the system is underactuated. The proposed solution is made up of two independent controllers. The first controller is developed for the rudder such that a course can be followed. The second controller is for the sail which will focus on maximizing speed while reducing the heeling angle and roll motion.

Multiple other control strategies for sailboats exists, from controllers based on simplified system dynamics (Jaulin and Bars [2013a], Clement [2013]), to fuzzy control (Stelzer et al. [2007], Yeh and bin [1992]) and neural networks (Tiano et al. [2001]). However, by not taking the full system dynamics into account the control strategies have to make up their inaccuracies by sub-optimal control. The rudder and sail controller developed in this thesis is not based on any previous work, though the course controller borrows an idea from [Xiao and Jouffroy, 2014] about a correction term in the rudder controller to make the sailboat able to follow a course instead of just a heading. A way of calculating this correction term is provided in this thesis.

Main contributions of chapter:

- A new course controller design is proposed, including a new method of easily calculating the correction term needed to sustain a given course.
- Optimal sail control which limits heeling angle and roll motion.
- Sail controller that takes into account the new and improved dynamics of the boom developed in the previous chapter.

4.1 Rudder Controller

4.1.1 Course Control

The objective of this controller is to get the χ (course angle) to the desired χ_d. A backstepping controller will be made to solve this problem, and a new solution is provided on how to effectively calculate the necessary drift angle for sailboats such that it is able to follow the course angle without the need for integral action.

Backstepping is a way of building up a controller using control Lyapunov functions (CLF) and a recursive design approach. This is done by first stabilizing one subsystem and then "backing out" to stabilize more of the system. This process is then repeated until the whole system is stable. The method only works for a very specific set of problems where one subsystem radiates out from another subsystem [Fossen, 2011].

new state variables

Lyapunov Functions

$$z_1 = \psi + \beta_b - \chi_d \qquad (4.1)$$

$$V_1 = \frac{1}{2} z_1^2 \qquad (4.2)$$

$$z_2 = r - \alpha \qquad (4.3)$$

$$V_2 = V_1 + \frac{1}{2} z_2^2 \qquad (4.4)$$

The first z-state describes the error from the desired course angle, and β_b is the necessary drift angle such that the desired course angle is achieved, referred to as the correction term (β_b is explained in more detail later on in this chapter). β_b is a bias, and its time derivative is equal to zero. The drift (also referred to as side slip) and course angle is defined as

$$\beta = arctan2(v \ cos(\phi), u), \qquad (4.5)$$

$$\chi = \psi + \beta \cdot \qquad (4.6)$$

The error dynamic of z_1 (which we will use later when solving the Lyapunov Function) is

$$\dot{z_1} = \dot{\psi} + \dot{\beta}_b - \dot{\chi}_d = z_2 + \alpha - \dot{\chi}_d \cdot \qquad (4.7)$$

Differentiating the first Lyapunov Function with respect to time gives

$$\begin{aligned} \dot{V_1} &= z_1 \dot{z}_1 \\ &= z_1(z_2 + \alpha - \dot{\chi}_d) \cdot \end{aligned} \qquad (4.8)$$

The goal of the virtual controller should be to make the Lyapunov Function stable,

$$\dot{V_1} = -K_P z_1^2 + z_1 z_2 \cdot \qquad (4.9)$$

It follows that the virtual controller is then equal to

$$\alpha = \dot{\chi}_d - K_P z_1 \tag{4.10}$$

Before differentiating the second Lyapunov function the error dynamics of z_2 is found to be

$$\dot{z}_2 = \dot{r} - \dot{\alpha}, \tag{4.11}$$

where $\dot{\alpha}$ is

$$\begin{aligned}\dot{\alpha} &= \ddot{\chi}_d - K_P \dot{z}_1 \\ &= \ddot{\chi}_d - K_P(z_2 + \alpha - \dot{\chi}_d) \\ &= \ddot{\chi}_d - K_P(z_2 + \dot{\chi}_d - K_P z_1 - \dot{\chi}_d) \\ &= \ddot{\chi}_d - K_P(z_2 - K_P z_1)\end{aligned} \tag{4.12}$$

Derivation of the second Lyapunov Function with respect to time can then be found:

$$\begin{aligned}\dot{V}_2 &= \dot{V}_1 + z_2 \dot{z}_2 \\ &= -K_P z_1^2 + z_1 z_2 + z_2(\dot{r} - \dot{\alpha}) \\ &= -K_P z_1^2 + z_2(\dot{r} - \dot{\alpha} + z_1)\cdot\end{aligned} \tag{4.13}$$

We then look to equation (3.2) to find $\dot{r}$. We construct the matrix C_ψ to be able to "extract" the elements from the vectors in equation (3.2) that are relevant:

$$C_\psi = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}, \tag{4.14}$$

$$\dot{r} = \frac{C_\psi(S + K + R - C_A(\nu_r)\nu_r - D(\nu_r))}{I_{zz} - N_{\dot{r}}}\cdot \tag{4.15}$$

Our actuator is the rudder, but we are going to set $u = R_\psi = C_\psi R$ and later see how to find σ such that $u = R_\psi$ is true. Setting $u = C_\psi R$ the equation becomes equal to

$$\dot{r} = \frac{C_\psi(S + K - C_A(\nu_r)\nu_r - D(\nu_r))}{I_{zz} - N_{\dot{r}}} + \frac{u}{I_{zz} - N_{\dot{r}}}\cdot \tag{4.16}$$

Inserting (4.16) into (4.13) we get

$$\dot{V}_2 = -K_P z_1^2 + z_2\left(\frac{C_\psi(S + K - C_A(\nu_r)\nu_r - D(\nu_r))}{I_{zz} - N_{\dot{r}}} + \frac{u}{I_{zz} - N_{\dot{r}}} - \dot{\alpha} + z_1\right)\cdot \tag{4.17}$$

The goal of the controller action u should be to make the Lyapunov Function stable, that is

$$\dot{V}_2 = -K_P z_1^2 - K_D z_2^2, \tag{4.18}$$

and it follows that u has to take the following form

$$\begin{aligned} u = &- C_{\psi}(S + K - C_A(\nu_r)\nu_r - D(\nu_r)) \\ &- (I_{zz} - N_{\dot{r}})(-\dot{\alpha} + z_1) - K_D z_2 \cdot \end{aligned} \tag{4.19}$$

Inserting (4.12) into (4.20) gives

$$\begin{aligned} u = &- C_{\psi}(S + K - C_A(\nu_r)\nu_r - D(\nu_r)) \\ &- (I_{zz} - N_{\dot{r}})(-\ddot{\chi}_d + K_P(z_2 - K_P z_1) + z_1 + K_D z_2) \cdot \end{aligned} \tag{4.20}$$

The correlation between σ and u has to be found, because it is the angle of the rudder which creates the desired moment. By expanding $u = R_{\psi}$, one finds that

$$u = R_{\psi} = l_4(R_L cos(\beta_{cr}) - R_D sin(\beta_{cr})) \cdot \tag{4.21}$$

Assuming that $v = p = r \approx 0$ it follows from (3.30) and (3.31) that $sin(\beta_{cr}) \approx 0$, which simplifies (4.21) to

$$u = R_{\psi} = l_4 R_L cos(\beta_{cr}) \cdot \tag{4.22}$$

The lift equation of a foil, (3.19), can easily be linearized for small angle of attacks ($|\alpha| < 15deg$). The linearized lift equation can be written as

$$F_{L0} = \frac{1}{2} pAC_{L0}\alpha V^2, \tag{4.23}$$

where C_{L0_r} is the linearized lift coefficient for the rudder. Inserting (4.23) into (4.22) we have

$$\begin{aligned} u &= l_4 R_L cos(\beta_{B_c r}) \\ &= l_4 \frac{1}{2} p_w A_r C_{L0_r} \alpha_r V_{cr}^2 cos(\beta_{cr}) \\ &= l_4 \frac{1}{2} p_w A_r C_{L0_r}(-\beta_{cr} + \sigma + \pi) V_{cr}^2 cos(\beta_{cr}), \end{aligned} \tag{4.24}$$

from which it follows

$$\sigma = \beta_{cr} - \pi + \frac{2u}{l_4 p_w A_r C_{L0_r} V_{cr}^2 cos(\beta_{cr})} \cdot \tag{4.25}$$

When calculating z_1 one would usually do the following procedure:

$$z_1 = mod(z_1 + \pi, 2\pi) - \pi, \tag{4.26}$$

where mod is the modulo operation. This ensures that the boat always turns in the direction that has the smallest error. This does however not work in this situation, because we want to do both the coming about and the jibing maneuver. If one calculates z_1 as in equation (4.26) one will always end up doing the tacking when sailing up-wind and jibing when sailing down-wind. How to select χ_d and how to choose between the two maneuvers will be handled by the path following algorithm in the next chapter.

4.1.2 Correction Term, β_b

As discussed, a heading controller is not good enough in the case of sailboats due to the fact that some side slip is needed to keep a course angle. However, using the course angle as feedback is troublesome because it is difficult to estimate. The idea of a correction term comes from Xiao and Jouffroy [2014], where they addressed the problem by using a nonlinear system solver to calculate the necessary drift angles to keep a desired course, then storing the results in a lookup table. The drawback of this is that the lookup table is quite large as it depends on several variables, and would need to be recomputed every time a small change is applied to the boat. In this thesis a much simpler way of calculating the correction term is provided.

The z_1 state for a traditional heading controller is equal to

$$\begin{aligned} z_1 &= \psi - \psi_d \\ &= \psi - (\chi_d - \beta_d) \\ &= \psi + \beta_d - \chi_d. \end{aligned} \tag{4.27}$$

On a sailboat, the keel causes hydrodynamic forces with the intent of counteracting undesirable forces. The undesirable forces are all the forces that cause the ship to move perpendicular to the course angle. In the body frame, these are all the forces created perpendicular to the drift angle. By assuming $\nu_r \approx \nu$ and $p = r \approx 0$, (3.26) simplifies to

$$V_{ck} = -\begin{bmatrix} u \\ v \end{bmatrix} = -\nu_{uv}. \tag{4.28}$$

Using (4.5), the simplification $v \ll u$ (low drift angle), and the approximation $tan(x) = x$ for small x, it follows that β_{ck} (3.27) can be computed by

$$\beta_{ck} = arctan2(V_{ck_v}, V_{ck_u}) = \frac{\beta}{cos(\phi)} + \pi \tag{4.29}$$

and that

$$\alpha_k = -\beta_{ck} + \pi = -\frac{\beta}{cos(\phi)}. \tag{4.30}$$

Assuming $cos(\phi) \approx 1$ and by applying the result from (4.29) into (3.29) we find that the lift of the keel is approximately perpendicular to the drift angle when either the heeling angle or drift angle is low. It then follows that the lift force from the keel is aligned with the undesirable force, because it is also perpendicular to the drift angle.

Defining the undesirable force, hereafter referred to as F_U, as -90 degrees to the drift angle causes the lift force generated by the keel and the unwanted forces to be aligned, but in opposite direction. Neither the restoring force or the Coriolis terms (assuming $r \approx 0$) contribute to the undesirable force. Putting it together we find

$$\begin{aligned} F_U =&(S_{x'} + R_{x'} - D_{x'}(\nu_r))cos(\beta - \frac{\pi}{2}) \\ &+(S_{y'} + R_{y'} - D_{y'}(\nu_r))sin(\beta - \frac{\pi}{2})\cdot \end{aligned} \tag{4.31}$$

The lift caused by the keel should be equal to the undesirable force. Linearizing the lift formula around an angle of attack of zero (as we did for the rudder), where C_{L0_k} is the linearized lift coefficient for the keel, we find the necessary drift angle:

$$\begin{aligned} F_U &= K_L \\ &= \frac{1}{2} p_w A_k C_{L0_k} \alpha_k V_{ck}^2 \\ \Rightarrow \alpha_k &= \frac{2F_U}{p_w A_k C_{L0_k} V_{ck}^2}\cdot \end{aligned} \tag{4.32}$$

From (4.30), it follows that

$$\beta_b = -\alpha_k cos(\phi)\cdot \tag{4.33}$$

However, β_b needs to be constrained due do practical reasons. If V_{ck} is very low then β_b will go towards infinity. Furthermore, the linearized lift approximation is only valid for small α_k. By constraining the solution to be within $\pm 10deg$ both problems are solved. An exception should also be put in place to avoid any numerical errors when $V_{ck} = 0$.

4.1.3 Turning Rate

The lift of the rudder is limited by the size of ν_r, which also limits the the maximum moment it can create in yaw. Unfortunately, the controller does not take this into account and could easily request more momentum than the rudder is able to handle, causing overshoot or other unexpected behavior. Furthermore, a step change in χ_d will cause the ship to turn in an unnatural manner because the yaw sub-system is tuned for a "mass-damper-spring" response, not a steady turning rate. Model predictive control (MPC) could potentially be applied to solve this problem, but a simpler approach has been chosen instead which is to limit the turning rate, such that $|\dot{\chi_d}| = r_{sat}$. A second order low-pass filter was added to make the signal smoother, which has the transfer function of

$$F(s) = \frac{\omega^2}{s^2 + 2\omega + \omega^2}, \tag{4.34}$$

where $T_\psi = \frac{1}{\omega}$ is treated as a tuning parameter.

4.2 Sail

4.2.1 Optimal Angle of Sail

The sail is used to create forward propulsion for the sailboat. There is an optimal sail angle that gives the highest forward acceleration for a given relative wind direction, effectively making λ_d a function of β_{ws}, $\lambda_d(\beta_{ws})$. The objective is thus to create a map from relative wind direction to the optimal angle of the sail. We begin by defining a way of measuring the force created by the sail in the forward direction independent of the wind speed:

$$S_{x'_r}(\beta_{ws}, \lambda) = \frac{S_{x'}(\beta_{ws}, \lambda, V_{ws})}{V_{ws}^2}, \tag{4.35}$$

where $S_{x'}$ is the force created by the sail in positive surge direction (see (3.25)) and $S_{x'_r}$ is the force divided by the wind speed (as experienced by the boat) squared. $S_{x'_r}$ is a function of λ and β_{ws}, and there will be an optimal λ given a β_{ws}.

However, λ cannot be chosen freely; it is restricted to a certain range depending on the wind direction and λ_{sat}. This is because we only have control of a rope limiting $|\lambda|$, and the rope only works through tension. When the wind is hitting from starboard, it follows that $\lambda > 0$ and that the torque created by the sail around the mast has to be positive (meaning a positive angle of attack). When the wind is hitting from port side, $\lambda > 0$ and the torque has to be negative (negative angle of attack).

The following equations describe the upper and lower limits of a stable angle of the sail, $\lambda \in [\lambda_l, \lambda_u]$, given a relative wind direction:

$$\lambda_u = \begin{cases} \beta_{ws} + \pi & \text{if } \beta_{ws} < 0 \\ 0 & \text{if } \beta_{ws} > 0 \end{cases} \tag{4.36}$$

$$\lambda_l = \begin{cases} 0 & \text{if } \beta_{ws} < 0 \\ \beta_{ws} - \pi & \text{if } \beta_{ws} > 0 \end{cases}. \tag{4.37}$$

In addition to (4.36) and (4.37), λ_{sat} still applies. That is, $\lambda_u \in [0, \lambda_{sat}]$ and $\lambda_l \in [-\lambda_{sat}, 0]$.

The optimal angle of the sail can then be found by traversal of all viable λ, trying to maximize $S_{x'_r}$ for a given β_{ws}. This can be computed off-line, and the results stored in a lookup table. In this paper, an optimal λ was found for a step size of $1deg$ in β_{ws}. In order to avoid unnecessary discontinuities in the control action, when there is only a small difference in the optimal λs between a step change in the lookup table, a new optimal λ is found based on a linearized solution between the two data points (this is computed on-line).

As discussed earlier, some drift is to be expected due to side forces and currents. The course angle is not equal to the heading angle, and it follows that the sail should be optimized for maximum speed in the desired course direction, or in body frame, the drift direction. The brute force approach to this problem is to find an optimal λ for each pair of β_{ws} and β, though this would make the lookup-table considerable larger.

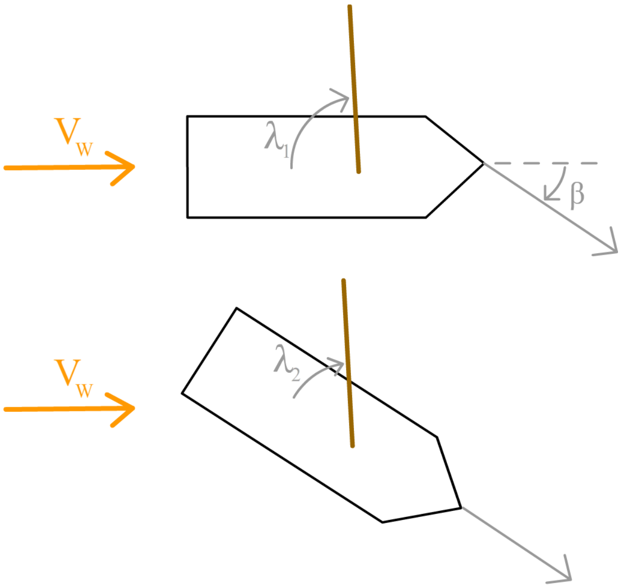


Figure 4.1: Optimal λ in different settings

Figure 4.1 shows one sailboat with a drift angle, and another sailboat with no drift angle, but they are both heading in the same direction. The angle of attack of the sail should be the same in both situations, and the following relation can be seen assuming a small heeling angle:

$$\lambda_1 = \lambda_2 + \beta, \qquad (4.38) \qquad\qquad \beta_{ws_2} = \beta_{ws_1} - \beta\cdot \qquad (4.39)$$

The lookup-table gives us the answer of optimal λ_2 when β_{ws_2} is found by (4.39), and λ_1 can then be calculated using (4.38). The correlations described in (4.38) do not take into consideration the limitations described in (4.36) and (4.37), but for small drift angles one can assume that thees play no effect.

4.2.2 Relative Momentum

In the same manner that we defined the relative force in surge direction, the relative momentum in roll is defined as

$$S_{\phi_r}(\beta_{ws}, \lambda) = \frac{S_\phi(\beta_{ws}, \lambda, V_{ws})}{V_{ws}^2}, \qquad (4.40)$$

where $S_{\phi r}$ is calculated at each optimal λ and is added to the lookup table. This solution of λ will hereafter, be referred to as the unconstrained solution, λ_{100}.

Before applying the control law to reduce roll motion and heel angle, we have to be able to control the amount of momentum created in roll. This is accomplished by finding new optimal λs, though with a restriction on the maximum relative momentum created at each solution.

To produce the desired result, the same traversal algorithm as discussed earlier is performed again but with the restriction that the solution has to produce a relative moment of 67% or less compared to that of λ_{100}. This solution will hereafter be referred to as λ_{67}. Both λ_{67} and its relative momentum is added to the lookup table.

This is then repeated for a 33%- and a 0% (or as low as possible) constrained solution, referred to as λ_{33} and λ_0. However, a new constraint is added, which is that $sign(\lambda_{100} - \lambda_{67}) = sign(\lambda_{67} - \lambda_{33}) = sign(\lambda_{33} - \lambda_0)$. In more practical terms, this is to ensure that the different solutions all move the sail in the same direction compared to the previous solution. This is important when we try to find a continuous solution of the relative momentum in the next step.

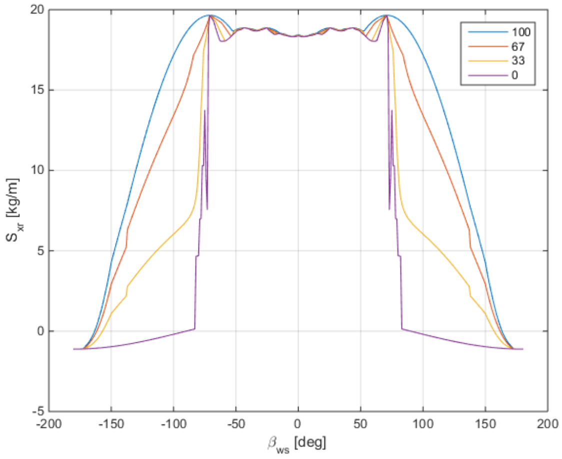


Figure 4.3: Relative force in surge caused by optimal sail position

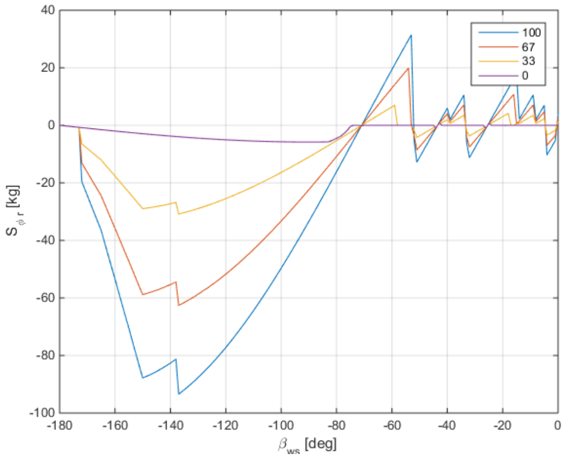


Figure 4.4: Relative momentum in roll caused by optimal sail position

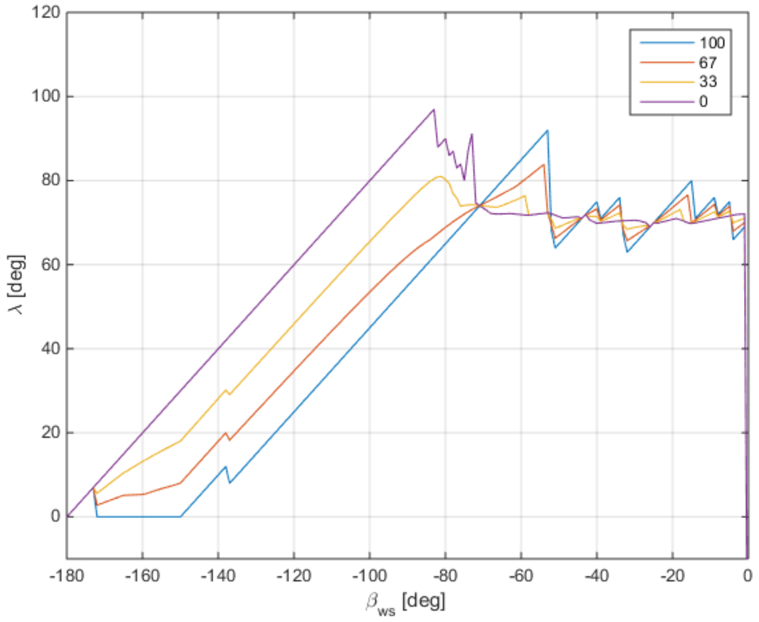


Figure 4.2: Optimal position of the sail

We then have all that we need to reverse the process. By entering β_{wr} one finds the solutions of λs that corresponds to that relative wind direction. Then, by applying the restriction of the desired $S_{\phi r}$, the ideal λ is found based on a linearization (computed on-line) between the different $S_{\phi r}$ created by λ_{100}, λ_{67}, λ_{33} and λ_0. The optimal sail angle can be looked upon like a function; $\lambda_d(\beta_{ws}, \beta, S_{\phi r_d})$, where $S_{\phi r_d}$ is the desired relative momentum.

4.2.3 Roll Controller

The roll controller will step in when the heeling angle gets too large. The heeling angle should not be larger than 10 degrees for a comfortable ride, and the desired heel angle, ϕ_d, is thus set equal to $\pm 10deg$ depending on the sign of ϕ, $\phi_d = sign(\phi)10deg$. Furthermore, roll motion should also be dampened, which gives the following error states:

$$e_1 = \phi - \phi_d, \quad (4.41) \qquad e_2 = r\cdot \quad (4.42)$$

The controller design method chosen is state feedback linearization [Freund, 1973, Isidori, 1989] combined with a linear quadratic regulator (LQR). The first step is to make a linearized system of the roll motion by canceling any non-linearities. Similar to the course controller, we make a matrix C_ϕ and define it as

$$C_\phi = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}\cdot \tag{4.43}$$

The roll sub-dynamic can then be written as

$$\begin{aligned} \dot{\phi} &= r \\ \dot{r} &= \frac{u + C_\phi(-D(\nu_r) - g(\eta) + K(\eta, \nu, V_c) + R(\eta, \nu, V_c, \sigma))}{I_{xx} - K_{\dot{p}}}\cdot \end{aligned} \tag{4.44}$$

Using the following control law

$$\begin{aligned} u = &- C_\phi(-D(\nu_r) - g(\eta) + K(\eta, \nu, V_c) + R(\eta, \nu, V_c, \sigma)) \\ &- K_{LQR}\begin{bmatrix} e_1 \\ e_2 \end{bmatrix} - K_{is}\int_0^t e_1 d\tau, \end{aligned} \tag{4.45}$$

where

$$S_{\phi r_d} = \frac{u}{V_{ws}^2}, \tag{4.46}$$

and K_{LQR} is the controller gain, it is easy to verify that all the non-linearity in the roll sub-dynamics are canceled, assuming $S_{\phi r_d} = S_{\phi r}$. Integral control action is added to remove errors caused by the difference in $S_{\phi r_d}$ and the actual $S_{\phi r}$, which is an effect of the linearization used to interpolate solutions of λ. Using conventional LQR theory, it follows that

$$x = \begin{bmatrix} \phi \\ p \end{bmatrix}, \quad (4.47) \qquad \dot{x} = Ax + Bu, \quad (4.48)$$

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \qquad (4.49) \qquad\qquad B = \begin{bmatrix} 0 \\ \frac{1}{I_{xx} - K_{\dot{p}}} \end{bmatrix}. \qquad (4.50)$$

The roll controller should not be on all the time, and the "obvious" solution would be to keep the controller on while $|\phi| > |\phi_d|$. However, this would not work out because of two reasons. Firstly, the controller would turn on and off all the time as the controller task is to have $|\phi| = |\phi_d|$. Secondly, the integrator would only be able to grow. While it might seem a bit counterintuitive to actively control ϕ towards ϕ_d when $|\phi| < |\phi_d|$, it is important to realize that a low ϕ is a trade-off between speed and robustness, and by allowing an increase in ϕ one increases the speed of the sailboat by loosening the restrictions on the sail.

The controller should be able to turn on quickly if $|\phi|$ gets very large, but not turn on unnecessarily. This is somewhat contradictory, and some compromise between the two has to be found. The controller should not turn off unless $\lambda = \lambda_{100}$, i.e. the maximum moment is given, and $|\phi| < \phi_d$. Using this logic, the following rules are proposed:

$$\dot{\Omega} = sign(e_1)e_1^2 K_c, \quad sat(|\Omega|) = 1 \qquad (4.51)$$

$$\Gamma = \begin{cases} 0 & \text{if } \Gamma = 1 \text{ and } \Omega = -1 \\ 1 & \text{if } \Gamma = -1 \text{ and } \Omega = 1 \\ \Gamma & \text{else} \end{cases} \qquad (4.52)$$

$$K_c = \begin{cases} K_{c_+} & \text{if } \Gamma = 0 \\ K_{c_-} & \text{if } \Gamma = 1 \text{ and } \lambda_d = \lambda_{100} \\ 0 & \text{else} \end{cases} \qquad (4.53)$$

$\Gamma = 1$ signals that the controller is turned on while $\Gamma = 0$ means that it is turned off. If the controller is off and $\Omega = 1$ then the controller should turn on. If the controller is on and $\Omega = -1$ the controller should turn off. Histories ensures that the controller do not turn on and off to fast, as this could potentially make the system unstable. The square of the error is used to make sure that the controller is able to turn on quickly in case $|\phi|$ gets very large, without triggering to easily if $|\phi|$ is barely larger than $|\phi_d|$ for a short period of time. Furthermore, if the sail can not create any more momentum in roll, that is $\lambda = \lambda_{100}$, the integral action is disabled such that unnecessary integral build up is avoided.

4.2.4 Rope Controller

The objective of the this controller is to control the length of the rope connected to the boom, l_{7_0}, such that the sail moves into the desired position (λ_d) given by the roll controller or the look-up table. Assuming that the rope has no stretch, l_{7_0} is equal to l_7, which is given by (3.47),(3.48) and (3.40). Replacing λ with λ_d and inserting (3.47) and (3.48) into (3.40) gives l_{7_0} as

$$l_{7_0} = \sqrt{(l_5 - l_6 cos(\lambda_d))^2 + (l_6 sin(\lambda_d))^2}. \tag{4.54}$$

Using (4.54) as an baseline and treating the stretching of the rope as an unknown error (which is slowly changing), the following controller action can be used for the actuator that is controlling the length of the rope

$$l_{7_0} = l_{7_b} - sign(\lambda) K_{ib} \int_0^t \lambda - \lambda_d d\tau, \tag{4.55}$$

where l_{7_b} is the baseline estimate of l_{7_0} assuming no stretch as given by (4.54) and K_{ib} is the integrator gain. The added complexity of the boom and sail modeling also introduces a couple of new control problems. The map presented in Figure 4.2 shows that the optimal solutions are not continuously, especially at $B_{ws} = 0$ where the step change in λ_d is about $140deg$. Wind gusts (or other disturbances, such as waves) could make the boat continuously change between the two optimal λ_ds, causing sub-optimal performance. To solve this problem, a special rule is put in place for this big jump. The rule is that if λ_d has to switch sign, the gain in $S_{x'_r}$, (4.35), has to be greater than 5%. Hysteresis was also considered, though implementing the method of using $S_{x'_r}$ is simpler and more flexible.

A second problem that arises is best explained by Figure 4.5. The current λ is shown by the brown boom, and the desired λ_d is shown by the dark blue boom. Using 4.55 directly as our control strategy, the boom will end up moving to the light blue position due to the wind pushing the sail in this direction. The solution is to first set $\lambda_d = 0$ such that the wind is hitting the sail from the correct side, then, when this position is reached, move the sail to the actual desired angle by loosening the rope again. The following rule is used, where λ_{dn} is the new and corrected λ_d:

$$\lambda_{dn} = \begin{cases} \lambda_d & \text{if } sign(\lambda) = sign(\lambda_d) \\ 0 & \text{else} \end{cases}, \tag{4.56}$$

The final problem is when $\lambda_d = 0$. Looking at (3.39) it becomes obvious that the force L_b goes towards zero as λ approaches zero, and λ will never actually get to zero unless the wind direction allows it. This can create integral build up, which should be avoid. The minimum length of the rope should be equal to $l_{min} = l_5 - l_6$, and when this happens the integral action should be turned off as the rope can not get any shorter.

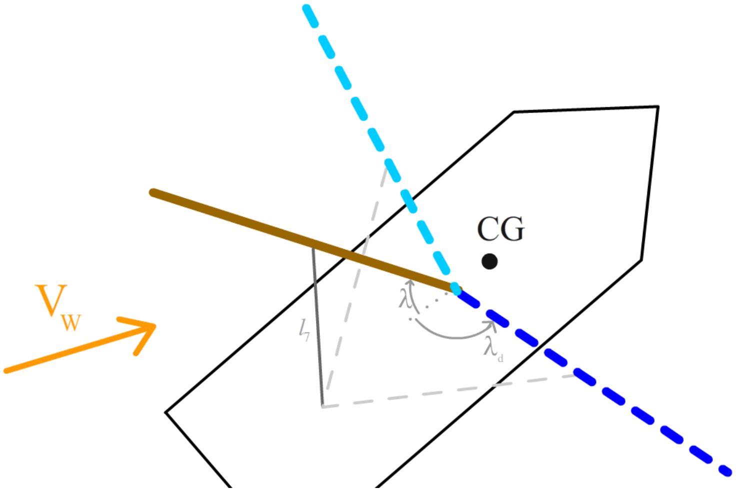


Figure 4.5: Illustration of angle of the sail control problem

4.3 Tuning of controllers

4.3.1 Course controller

The course controller was tuned manually. The controller gains K_p and K_d where tuned for what could be considered a normal speed in surge, $u = 4\frac{m}{s}$. The reference gain T_ϕ was tuned until the beginning and end of the turn looked smooth. Then, r_{sat} was tuned by lowering the speed in surge to $u = 2\frac{m}{s}$, and r_{sat} was lowered until there were no more overshoot. The results are presented in Table 4.1.

tuning parameter	**value**
K_p	0.25
K_d	50
T_ψ	0.66
r_{sat}	$15\frac{deg}{s}$

Table 4.1: Course controller tuning parameters

4.3.2 Roll controller

The LQR controller gain, K_{LQR}, was tuned by setting the matrices Q and R, where Q is the weighting matrix in the quadratic cost caused by error in x, and R weighs the quadratic cost of using the actuator. R was tuned to be $R = 1e\text{-}10$ while Q was tuned to be

$$Q = \begin{bmatrix} 0.75 & 0 \\ 0 & 1 \end{bmatrix}. \tag{4.57}$$

The LQR gain was then calculated in matlab by using the function lqr(A,B,Q,R,0). K_{is}, K_{c_+} and K_{c_-} where tuned manually. The tuning parameters of the roll controller can be found in Table 4.2.

tuning parameter	**value**
K_{LQR}	[8.66e4 1.05e5]
K_{is}	2500
K_{c_+}	200
K_{c_-}	100

Table 4.2: Roll controller tuning parameters

4.3.3 Rope controller

K_{ib} was tuned manually. Value of K_{ib} is found in Table 4.3.

tuning parameter	**value**
K_{ib}	0.2

Table 4.3: Rope controller tuning parameters

4.4 Results

4.4.1 Course controller

Two sets of simulations were executed to show the effectiveness of the drift correction term. In the first simulation β_b is set equal to zero, meaning that the controller will behave as a heading controller. In the second simulation β_b is calculated properly. In both simulations, the boat will try to keep a steady course angle of 0 degrees (north) when the wind speed is $5\frac{m}{s}$ and the angle of the wind is equal to $\beta_w = 125deg$. The simulations lasted 200 seconds.

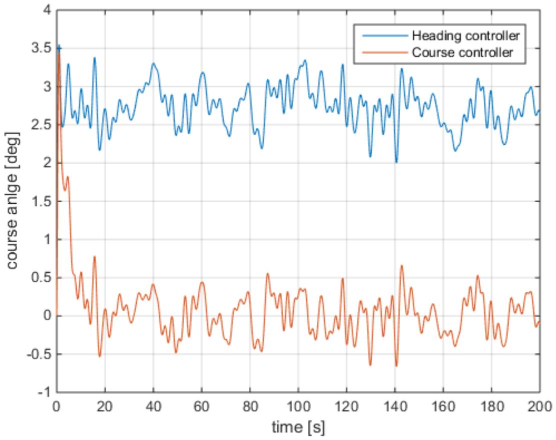


Figure 4.6: Course controller, time evolution of χ

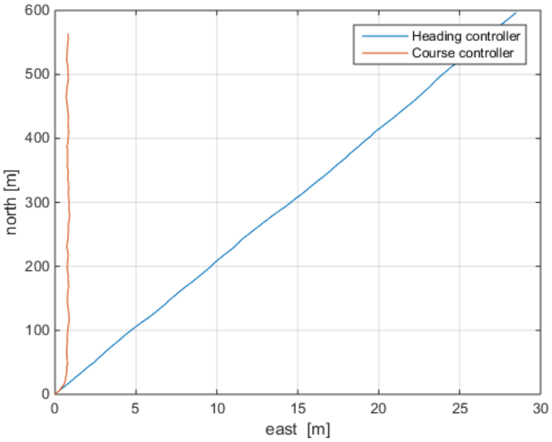


Figure 4.7: Course controller, North/East-plot

The overall drift from the desired course can be estimated by figure 4.7. To compute an estimate of the error, $|atan(\frac{x}{y})|$ is used, where x is the distance traveled in north direction and y is the distance traveled in east direction. The course controller takes 20 seconds to adjust, and the drift caused by this is removed from the error. See table 4.4 for results.

Table 4.4: Results of course controller test

simulation	x $[m]$	y $[m]$	error $[deg]$
heading controller	28	590	2.71
course controller	0	560	0

Comparing with figure 4.6, the estimate of the errors seems reasonable. The course controller managed to follow the desired course much better than the heading controller, an on average managed to follow the course perfectly. Looking at the results one can conclude that the course controller behaved as intended.

A third and fourth simulation was made to show how the control system handles turning at different speeds. The boat will try to do a $90deg$ turn at initial speeds of $u = 5\frac{m}{s}$ and $u = 3\frac{m}{s}$. The wind speed was set to zero.

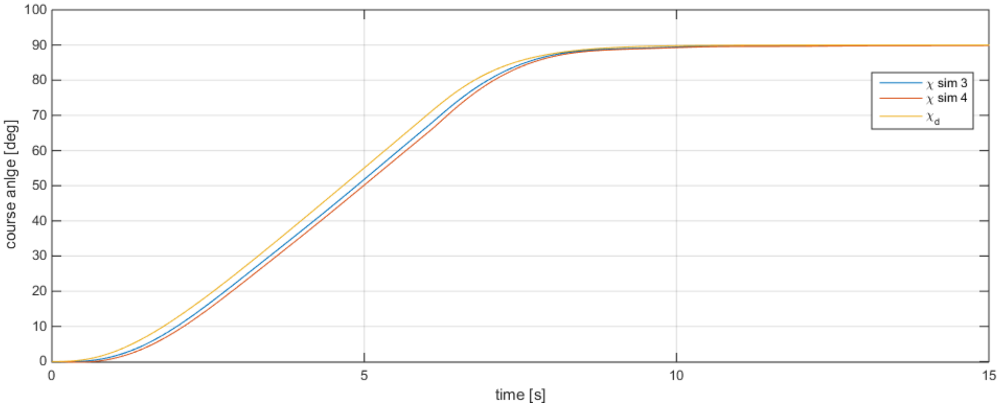


Figure 4.8: Course controller, time evolution of χ

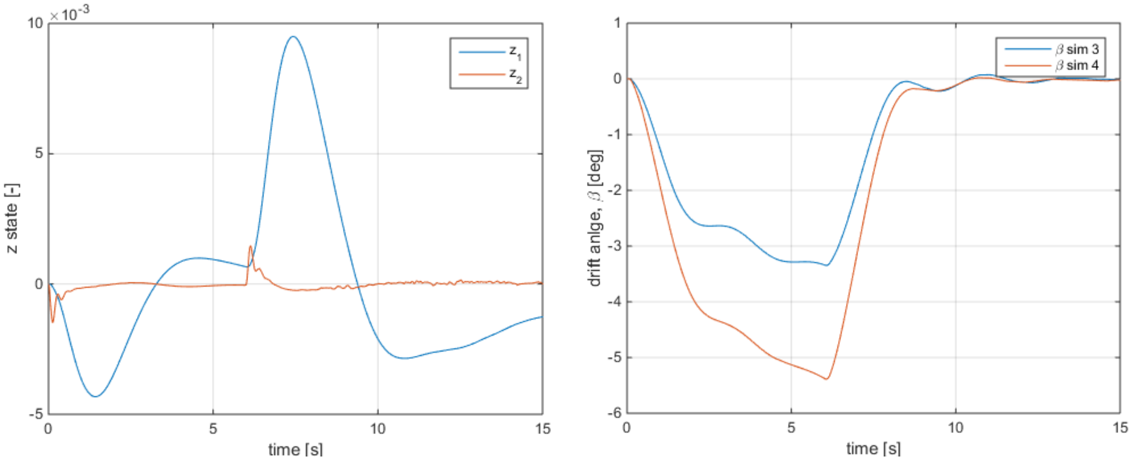


Figure 4.9: Course controller, time evolution of z-states

Figure 4.10: Course controller, time evolution of β

As can be seen from Figure 4.8, the control system is able to complete the turn. There is a small offset from the course, which is due to the drift angle. The drift angle increases as the speed in surge drops. The course of the two simulations and the desired course are very similar, to the point where they are difficult to distinguish for each other. The turning rate is steady, and even at low speeds we get not overshoot. The speed in surge at $t = 10s$ were $u = 2.97\frac{m}{s}$ and $2.05\frac{m}{s}$ respectively, meaning that a considerable speed was lost during the turn. Figure 4.9 shows the time evolution of z_1 and z_2 of the third simulation. The terms are very small and behaves as expected.

4.4.2 Roll controller

Two sets of simulations have been studied in what follows. In both simulations the boat is set to go on a course direction perpendicular to a mean wind speed of $10\frac{m}{s}$. The results will show the last 50 seconds of a 200 second simulation, ensuring that the top speed in surge has been reached and that the integral term in the controller has had time to reach a steady value. In the first simulation, the controller in roll has been turned off, meaning λ_{100} is always chosen as the desired angle of the sail. In the next simulation the controller is turned on with $\phi_d = 10deg$.

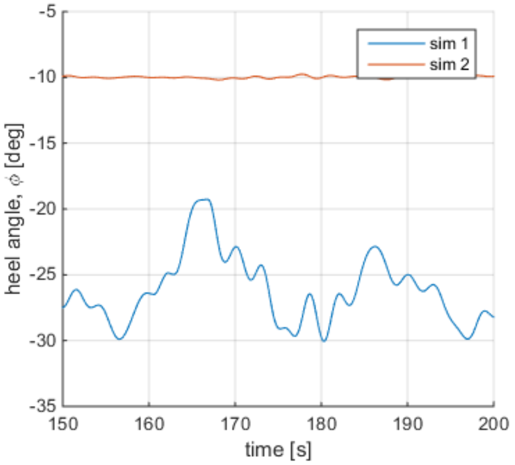


Figure 4.11: Roll controller, time evolution of ϕ

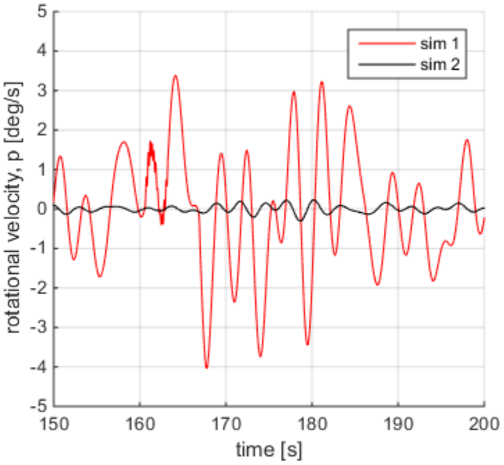


Figure 4.12: Roll controller, time evolution of p

Figure 4.11 demonstrates the effectiveness of the controller keeping the heel angle low. Furthermore, not only does the controller reduce the maximum heeling angle, it also reduces the amplitude of the roll motion almost completely. Figure 4.12 shows that the rotational velocity is reduced considerably. The results of the test are found in Table 7.2. The amplitude of the roll motion is reduced by 95.5% and that the maximum rotational velocity is reduced by by 92.5%. However, the reduction of the heeling angle comes at a price; the speed in surge is reduced by 18.6%.

Table 4.5: Results of roll controller test

simulation	amplitude of ϕ $[deg]$	maximum p $[\frac{deg}{s}]$	u $[\frac{m}{s}]$
controller off	5.31	4.0	7.58
controller on	0.24	0.3	6.48

While it in general is true that loosening the restriction on ϕ increases the speed in surge, this is only true until a certain point. As the heeling angle gets very large the effective wind direction approaches $-\pi$, i.e. equivalent to sailing directly up-wind. This can bee seen from (3.22) and (3.23). The following result shows the maximum speed in surge when targeting different heeling angles with the roll controller.

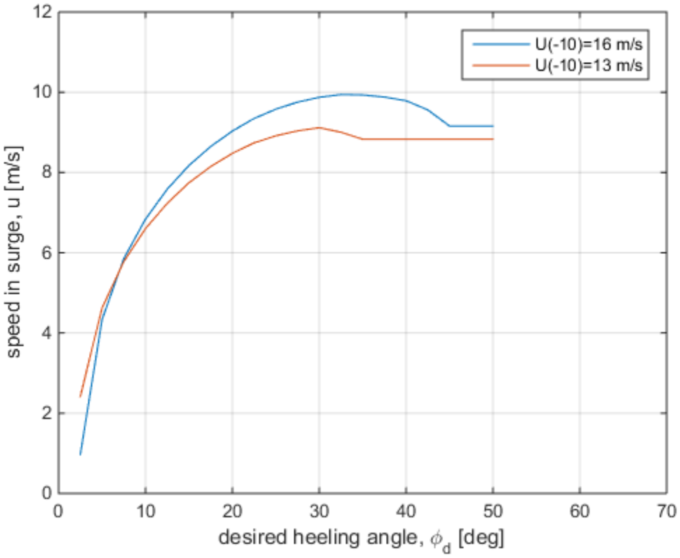


Figure 4.13: Relation between heeling angle and speed in surge

Figure 4.13 one can see the results of the simulations. After a while the speed does not change as ϕ_d increases. This is due to λ_{100} already being chosen, and the sail is not able to produce any more momentum (without choosing a non-optimal sail angle). There is clearly an optimal heeling angle for forward speed, and it seems to be moving up when the wind speed increases. At a wind speed of $16\frac{m}{s}$ the optimal heeling angle is $32.5deg$, and at $13\frac{m}{s}$ wind speed the optimal angle is $30deg$. From the looks of it, restricting the heeling angle would make even more sense for even higher wind speeds as the ratio between maximum speed at optimal heeling angle and maximum speed at unrestricted heeling angle increases by wind speed. The optimal heeling angle in terms of speed is quite high, and is way above the "recommended" 10 degrees. However, these results might be interesting for robotic racing applications.

4.4.3 Rope controller

The simulation shown recreates the situation described in Figure 4.5. $\beta_w = 20deg$, wind speed of $5\frac{m}{s}$, $\psi = 0$, $\lambda = 90deg$ at the beginning, and $\lambda_d = -90deg$. $\dot{\nu} = \nu = 0$ to ensure that the boat is standing still, and wind gusts and the wind direction changing are disabled.

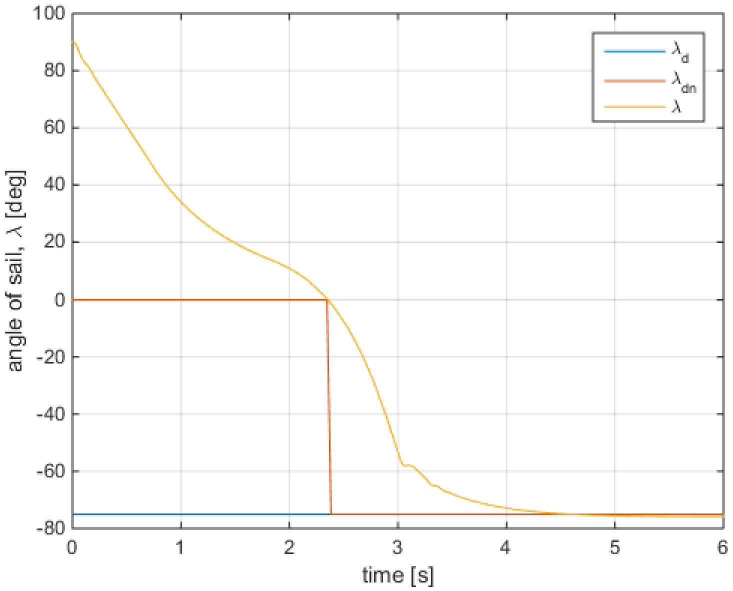


Figure 4.14: Rope controller, λ-plot

From Figure 4.14 one can see that the controller works as intended. When $\psi = \phi = 0$ and $\nu = 0$ it follows that $\beta_{ws} = \beta_w$, and comparing λ_d in Figure 4.14 to the optimal lambda in Figure 4.2 one can see that the optimal sail angle is chosen correctly. λ_{dn} is set to zero at the beginning because of (4.56), and is then correctly set equal to λ_d when $sign(\lambda) = sign(\lambda_d)$. At $t = 3$ there is a small bump in λ before it stabilizes at λ_d. This is because the rope that controls the boom has not yet been able to reach it correct length due to the limitations of the rope actuator (modeled as a 1. order low pass filter).

5 Path Following

Path following and path generation is an important aspect of most autonomous vehicles, but it is especially important for sailboats due to the limitations of the sail and rudder. For most vehicles the shortest path in air distance is equal to the path that takes the shortest amount of time to travel. This is not true for sailboats, the most obvious example being if we want to travel up-wind. This is a non-sailing zone which has to be avoided. To be able to sail up-wind the sailboat has to move in a zig-zag like pattern called beating. For sailing optimally down wind one also has to move in the same zig-zag pattern, because sailing broad reaching is faster than running.

In this chapter a relatively simple to implement method for path following is developed. The solution consists of a supervisor deciding if the path is up-wind, nominal or down-wind, and then applying the correct control strategy depending on the situation. The algorithm is also able to chose the most appropriate turning maneuver (tack or jibe) depending on the wind and the speed of the ship. The optimal turning maneuver should be the one that makes the shortest path without compromising on safety or the risk of loosing control over the ship.

A Lookahead-based steering algorithm (Fossen [2011],pp. 254-266) will be the basis of the point-to-point following algorithm in nominal mode, but it has to be modified to be able to solve the non-sailing zone problem. The modifications are inspired by Jaulin and Bars [2013a], and this thesis expand upon their work by increasing precision and robustness. The issue of choosing optimal turning maneuver have already been explored by Xiao and Jouffroy [2011], and the reasoning behind the strategy developed in this thesis will be similar, though less complex.

Main contributions of chapter:

- Path following, enabling sailing up-wind and down-wind optimally.

- An easy to implement method of choosing between tacking or jibing.

5.1 Polar Speed Diagram

The Polar speed diagram describes the speed of the sailboat when traveling in a given direction relative to the wind. This is valuable to known when deciding how steep one should travel relative to the wind direction.

While there are ways of estimating the polar speed diagram of a sailboat using the dynamics established in chapter 3 (Herrero et al. [2005]), a more brute force method of computing the polar speed diagram is used in this thesis. Multiple simulations were set up where the boat had to travel in a straight line with different course angles relative to the wind. The polar speed diagram were then calculated at 2,4,6,8,10 and 12 $\frac{m}{s}$ wind and one value was computed pr 5 degree of course angle. Wind guests were not enabled. Each simulation lasted 200 seconds to make sure the boat had reached a steady state.

To make the process a bit easier an automated script was made in matlab. Calculating all the values took roughly 2 hours on an i5-4670k (3.4GHz). The script prints out an ETA of the calculation, average simulation time and percentage completed every two minutes such that one can follow the progress. Figure (5.1) shows the result, where 0 degrees would be sailing directly against the wind.

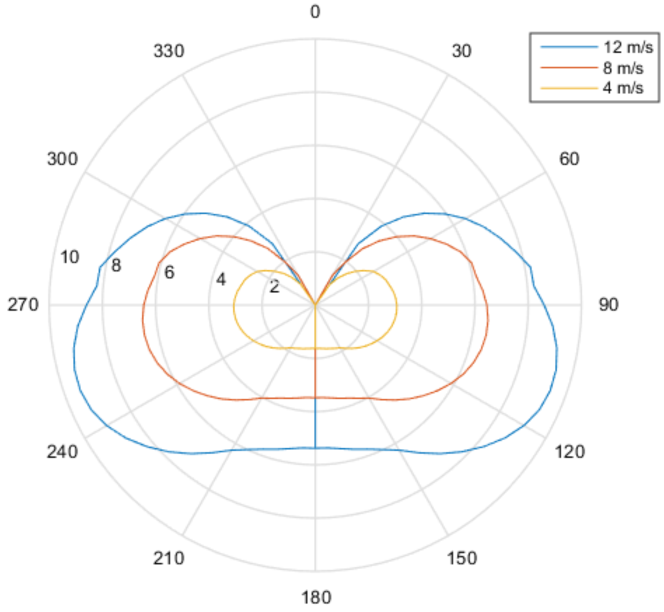


Figure 5.1: Polar speed diagram

Θmax is the optimal direction relative to the wind direction when sailing down-wind, and Θmin the optimal angle when sailing up-wind. Assuming that the polar speed diagram is symmetrical along, Θmax and Θmin is given by the angle that maximizes the following expressions

$$U_{uw} = cos(\Theta max)U(\Theta max) \quad (5.1) \qquad U_{dw} = -cos(\Theta min)U(\Theta min), \quad (5.2)$$

where $U(\Theta)$ is the speed of the ship in a given direction, and U_{uw} and U_{dw} is the effective maximum speed when traveling up-wind and down-wind. Using the data from the polar speed diagram, Θmax and Θmin were calculated at at the different wind speeds, and the results are presented in Table 5.1. As can be seen, Θmax and Θmin are fairly consistent regardless of the wind speed. From now on, Θmax and Θmin is given by the $8\frac{m}{s}$ wind speed data, i.e. $\Theta max = 140deg$ and $\Theta min = 55deg$.

Wind speed $[\frac{m}{s}]$	Θmax $[deg]$	Θmin $[deg]$
4	140	55
8	140	55
12	140	50

Table 5.1: Optimal course angle for up-wind and down-wind navigation

5.2 Path Following

5.2.1 Lookahead-based steering

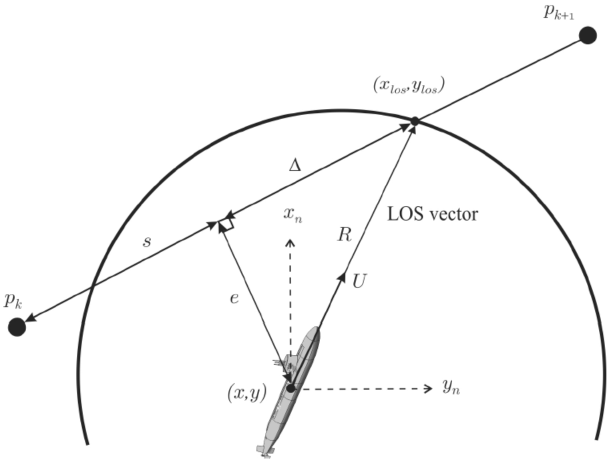


Figure 5.2: Lookahead-based steering (Fossen [2011], p. 262)

Lookahead-based steering is the basis for the path following algorithm (Fossen [2011],pp.254-266). Utilizing this method, the desired course angle is given by

$$\chi_d(e) = \chi_p + \chi_r(e), \tag{5.3}$$

where χ_p is the angle of the path,

$$\chi_p = \alpha_k = atan2(y_{k+1} - y_k, x_{k+1} - x_k)\cdot \tag{5.4}$$

x_k and y_k is the coordinates of point k along the path. χ_r ensures that the boat is heading towards a point on the path that is located at the lokahead distance Δ from the ship

$$\chi_r(e) = arctan2(-e, \Delta), \tag{5.5}$$

where e is the cross track error

$$e = -(x - x_k)sin(\alpha) + (y - y_k)cos(\alpha)\cdot \tag{5.6}$$

The lookahead distance, Δ, is given by

$$\Delta = \sqrt[2]{R^2 - e^2} \tag{5.7}$$

where R is the selected "circle of acceptance", see figure 5.2. A large R creates a long lookahead distance, while a small R makes for more aggressive steering. A lower bound of 0 is set on Δ because a negative (/complex) Δ do not make sense.

k will increase when the distance between p_{k+1} and ν_{xy} is less than R_k, or when the sailboat has reached the end of a the line between p_k and p_{k+1}, that is $\langle p_{k+1} - p_k,\ \eta_{xy} - p_{k+1}\rangle > 0$. When the last point has been reached the simulation will stop. It was found that basing R_k and R on the turning radius gave good results, which is roughly equal to

$$R_t(u) = \frac{|u|}{r_{sat}} \tag{5.8}$$

by testing it was shown that $R_k = R_t(u)$ and $R = 5R_t(u)$ were effective.

5.2.2 Sailing up- and down-wind

The Lookahead-based steering algorithm do not take into account the fact that the sailboat can not sail to steep against the wind. The method used in this thesis follows a similar structure to Jaulin and Bars [2013a], whereby a supervisor decides between two modes: nominal route or tack. In nominal route mode the desired heading is equal to the angle between the two points p_k and p_{k+1}. In tack mode the desired heading angle is based off the wind orientation $\pm\Theta_{min}$, and a tack is initiated when the cross track error becomes to large. k increases when the sailboat crosses the line which is perpendicular to p_k and p_{k+1} and goes though point p_{k+1}.

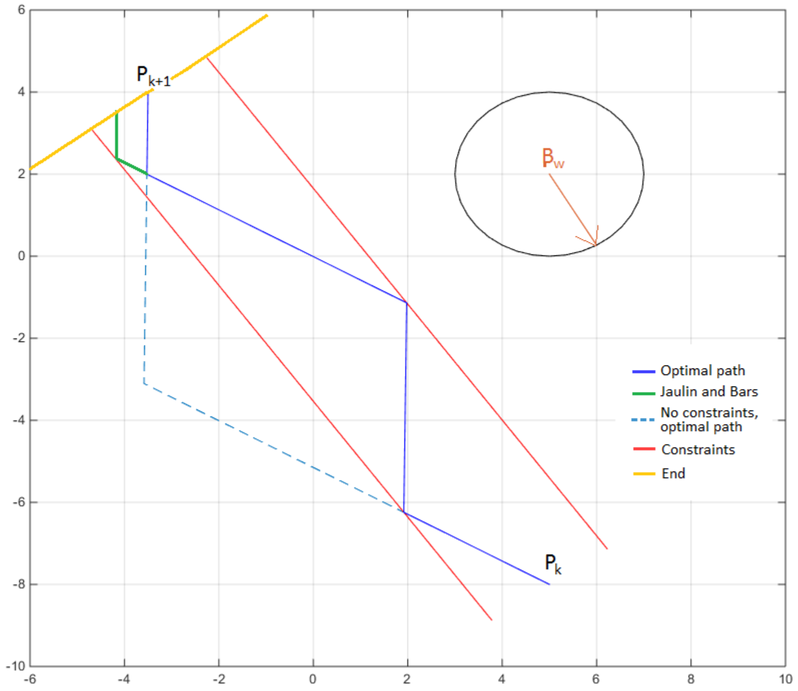


Figure 5.3: Up-wind sailing (beating) algorithm

The supervisor developed in this thesis build upon this work and adds another mode of operation for down-wind sailing.
Figure 5.3 shows how the up-wind sailing problem optimally should be handled, which is the dark blue line. The green line represents the algorithm developed by Jaulin and Bars [2013a], and how it would miss the point p_{k+1}. It is obvious that the error of the algorithm can be as big as the size of the constraint, but a small constraint would mean a lot of tack or jibe maneuvers. Ideally one would only tack or jibe one time, as shown by the light-blue line, though this might be impossible due to constraints in the environment (reefs, shore, etc).

A measurement for how steep one are sailing up against the wind is given by [1]

$$\kappa_p = \chi_p - \beta_w + \pi \cdot \tag{5.9}$$

When $\kappa_p = 0$, the path created by χ_p goes through the eye of the wind (up-wind). $|\kappa_p|$ should not be smaller than the optimal angle for sailing against the wind, Θ_{min}. If $|\kappa_p| < \Theta_{min}$, the up-wind mode should be enabled. Inserting χ_d for χ_p and $\pm\Theta_{min}$ for κ_p in (5.9) and solving for χ_d gives

[1] In equation (5.9), and the following equations, it is important to use modulus to keep the values valid by keeping them in the range of $\pm\pi$. It should be easy to understand from context which equations this entails.

$$\chi_d = \pm\Theta_{min} + \beta_w + \pi\cdot \tag{5.10}$$

Each time the sign in front of Θ_{min} is changed, a tack or jibe will be initiated. By the strategy of Jaulin and Bars [2013a], a turn should happen when the cross track error gets to big, $|e| > e_{max}$. By slightly modifying this to $|e| + R_t(u) > e_{max}$ the tack or jibe will be initiated early enough to avoid $|e| > e_{max}$. The sign in front of Θ_{min} should change to the opposite sign of e, ensuring that the course changes towards the line between p_k and p_{k+1} (towards the path). The angle between p_{k+1} and the sailboat is given by

$$\chi_s = arctan2(y_{k+1} - y, x_{k+1} - x), \tag{5.11}$$

where y and x is the position of the boat. A second reason for turning is to avoid the large error seen in Figure 5.3, and should be when the sailboat can sail directly against P_{k+1} without any problems. Defining κ_s as

$$\kappa_s = \chi_s - \beta_w + \pi, \tag{5.12}$$

it follows that a tack should be initiated when $|\kappa_s| > \Theta_{min}$. Not only should the boat tack if this is true, but it should aim straight for P_{k+1}, that is $\chi_d = \chi_s$ when $|\kappa_s| > \Theta_{min}$.

The algorithm for sailing down-wind is similar to when sailing up-wind. When $\kappa_p = \pi$, the path is down-wind. If $|\kappa_p| > \Theta_{max}$, the down-wind mode should be enabled. When sailing down-wind the desired course angle is given by

$$\chi_d = \pm\Theta_{max} + \beta_w + \pi, \tag{5.13}$$

and the conditions for initiating a jibe and changing the sign in front of Θ_{max} are similar to when sailing up-wind: $|e| + R_t(u) > e_{max}$ or if $|\kappa_s| < \Theta_{max}$, and the sign in front of Θ_{max} should be equal to the sign of the cross track error (opposite to when sailing up-wind).

$$\kappa_{\chi d} = \chi_d - \beta_w + \pi \tag{5.14}$$

If $|\kappa_s| > \Theta_{min}$ and $|\kappa_s| < \Theta_{max}$ the nominal mode should be selected. The nominal mode uses the lookahead-based steering algorithm to primarily set χ_d. However, care has to be taken because $|\kappa_{\chi d}| < \Theta_{min}$ (5.14) is not unlikely, as shown in Figure 5.4 where even though the path itself is not up-wind the lookahead-based steering algorithm computes a desired heading which is up-wind. If $\kappa_{\chi d} < \Theta_{MIN}$, where $\Theta_{MIN} = 40deg$ (See Figure 5.1) is the steepest angle one can travel up wind at a reasonable speed, we should chose

$$\chi_d = \pm\Theta_{min} + \beta_w + \pi, \tag{5.15}$$

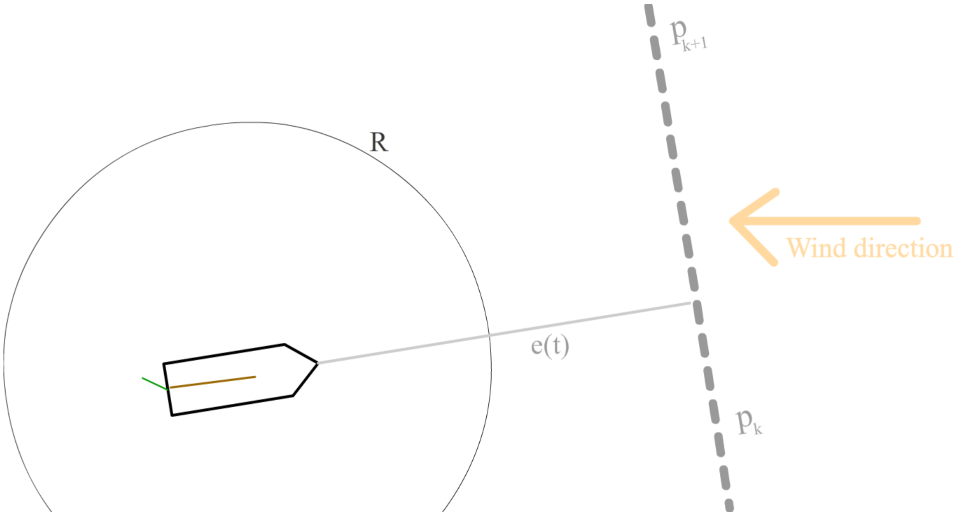


Figure 5.4: Nominal path following problem

where the sign in front of Θ_{min} should be set such that the cross track error gets smaller as fast as possible. If the two different options are χ_{d1} and χ_{d2} then the course angle that differs the most from χ_p (or $\chi_p + \pi$) will make the cross track error disappear the fastest, as the cross track error is measured perpendicular to χ_p. The following rules are proposed:

$$\chi_d = \begin{cases} \chi_{d1} & \text{if } |mod(\chi_p - \chi_{d1} + \frac{\pi}{2}, \pi) - \frac{\pi}{2}| > |mod(\chi_p - \chi_{d2} + \frac{\pi}{2}, \pi) - \frac{\pi}{2}| \\ \chi_{d2} & \text{else} \end{cases} \tag{5.16}$$

This algorithm takes over for the lookahead-based steering until $|e| - R(t) < 0|$. Hysteresis has been implemented to avoid the supervisor from changing back and forth between the different modes too fast due to a variable wind direction. A safety factor ϵ_Θ is proposed, such that if nominal mode is selected, $\epsilon_\Theta + |\kappa_p| < \Theta_{min}$ has to be true before switching to up-wind mode. If it wants to go back to nominal mode again then $|\kappa_p| > \Theta_{min} + \epsilon_\Theta$ has to be true. The same logic is applied to the down-wind sailing mode.

5.3 Tack or Jibe

When sailing up-wind one can choose between tacking or jibing to turn the ship. Tacking is considered more dangerous as one has to cross the no-sailing zone. During this time the sail will not be providing any forward force, and if the speed drops too much one can lose control. In the paper by Xiao and Jouffroy [2011] the analogy of a kid jumping across a river is used. Before the kid can jump he has to acquire enough speed, such that when his feet leaves the ground he has enough forward momentum to carry him over the river. The same idea is applicable for a sailboat crossing the no-sailing zone; there has to be enough initial energy in the system before a tack can be initiated, else it will fail. The surge sub-dynamic is given by

$$(m - X_{\dot{u}})\dot{u} - mrv + Y_{\dot{v}}vr + Y_{\dot{r}}r^2 + D_{qx}(u) + d_{l_{11}}u = S_{x'} + K_{x'} + R_{x'}\cdot \tag{5.17}$$

Assuming $u > 0$, $R_{x'} \approx 0$, $\alpha_s = \alpha_k \approx 0$, $r = sign(r)\ r_{sat}$, equation (5.17) simplifies to

$$(m - X_{\dot{u}})\dot{u} = -Y_{\dot{r}}r_{sat}^2 - D_{qx}(u) - d_{l_{11}}u - D_K u^2 - D_S V_{ws}^2 \pm m\ sign(r)\ r_{sat}v, \tag{5.18}$$

where D_S and D_K are quadratic drag constants for the sail and keel. For the relation $r = sign(r)\ r_{sat}$ to be true a minimum speed in surge of $u_{min} = 2\frac{m}{s}$ is required. The results from chapter 4 showed that while turning the drift would not get higher than $5deg$ as long as $u_{min} > 2\frac{m}{s}$. Using the approximation $U = \sqrt{u^2 + v^2} \approx u$ one can find an approximation of v:

$$v \approx u\ sin(-\beta_{max} sign(r)), \tag{5.19}$$

where $\beta_{max} = 5deg$, from which it follows that

$$mrv = m\ sign(r)\ r_{sat}\ u\ sin(-\beta_{max} sign(r)) = -m\ r_{sat}\ u\ sin(\beta_{max}) = -D_r u, \tag{5.20}$$

where $D_r = m\ r_{sat}\ sin(\beta_{max})$. Inserting (5.20) into (5.18) gives

$$(m - X_{\dot{u}})\dot{u} = -Y_{\dot{r}}r_{sat}^2 - D_{qx}(u) - (d_{l_{11}} + D_r)u - D_K u^2 - D_S V_{ws}^T V_{ws}, \tag{5.21}$$

which can be simplified by gathering some of the drag terms to

$$(m - X_{\dot{u}})\dot{u} = -Y_{\dot{r}}r_{sat}^2 - D_\gamma u^2 - D_\rho u - D_S V_{ws}^T V_{ws}, \tag{5.22}$$

where $D_\rho = d_{l_{11}} + D_r$ is a linear drag term for the linear drag of the hull and the rigid body coriolis term, and D_γ is a quadratic drag constant that accounts for the drag of the hull and keel. It follows that the following equation must hold:

$$D_\gamma u^2 > D_{qx}(u) + D_K u^2\cdot \tag{5.23}$$

Setting a minimum speed in surge, u_{min}, and inserting for the quadratic drag equation of the hull (3.14), and the drag equation for the keel at $\alpha_k = 0$ (which is simply the viscus drag, which dominates at small α_s), gives

$$D_\gamma = \frac{1}{2} p_w C_F(R_n(u_{min}))\Big(S(1+k) + 2A_K(1+2\frac{t_{max}}{c})\Big)\cdot \tag{5.24}$$

$C_F(R_n(u))$ decreases as u increases, and an upper bound of $C_F(R_n)$ can be calculated at $u = u_{min}$. Given $V_{wU} =\sqrt{V_w^T V_w}$, an upper bound estimate of V_{ws} can be expressed as

$$V_{ws}^T V_{ws} = (u + V_{wU})^2\cdot \tag{5.25}$$

The different coefficients where then calculated, and are presented in Table 5.2. D_S might look insignificant, though it is important to keep in mind that it is proportional to $(u + V_{wU})^2$ and will easily dominate in high wind speed situations. D_r makes up about 81% of D_ρ, meaning that the rigid body coriolis term is a very important part of the effective linear drag. In the paper by Xiao and Jouffroy [2011] the rigid body coriolis term is neglected, as well as D_S, even though both are clearly an important part of the surge sub-dynamic while tacking!

Coefficient	value
D_γ	18.0
D_ρ	44.7
D_S	1.11

Table 5.2: Surge sub-dynamic coefficients

The time it takes for a tack to complete while, and only taking into account when the sailboat looses speed, is roughly equal to

$$T = \frac{2\Theta_{MIN}}{r_{sat}}\cdot \tag{5.26}$$

The speed in surge should not be lower than u_{min} at the end of the turn, i.e. $u(t+T) > u_{min}$. Using the Euler Forward Method for integration, an estimate of $u(t+T)$ can be found given the initial speed in surge and the absolute wind speed (V_{wU}). The Euler Forward Method for integration is

$$u_{t+h} = u_t + h\ \dot{u}(u, V_w), \tag{5.27}$$

where h is the step size and

$$\dot{u}(u, V_w) = \frac{-Y_{\dot{r}} r_{sat}^2 - D_\gamma u^2 - D_\rho u - D_S(u + V_{wU})^2}{(m - X_{\dot{u}})}\cdot \tag{5.28}$$

The local truncation error of Forward Euler is given by

$$LTE = u(t+h) - u_{t+h} = \frac{1}{2} h^2 \ddot{u}(\zeta), \tag{5.29}$$

where $\zeta \in [t, t+h]$. As long as $-Y_{\dot{r}} r_{sat}^2 < D_\gamma u^2 + D_\rho u + D_S(u + V_{wU})^2$ then $\ddot{u} > 0$, which was found to be true when $u > 0.6\frac{m}{s}$. From (5.29), when $\ddot{u} > 0$ then $u(t+h) > u_{t+h}$. It then also follows that $u(t+T) > u_{t+T}$, meaning that the Euler Forward Method for integration in this case gives a lower bound estimate, and a relatively large step size h can be used without concern for underestimating the dampening. Now that we are able to optimally choose between tacking and jibing a method for setting χ_d correctly has to be found. Normally when creating a heading or a course control one use the following technique to make the boat turns the shortest direction:

$$z_1 = mod(\psi + \beta_b - \chi_d + \pi, 2\pi) - \pi\cdot \tag{5.30}$$

This does however not work in this case because the jibe maneuver is not the shortest path. Taking a step back, if there exist a point between χ_d and ψ ($\beta \approx 0$) which makes $\kappa_p = 0$ we have to cross the eye of the wind. Given

$$\kappa_\psi = \psi - \beta_w + \pi \tag{5.31}$$

if $sign(\kappa_\psi) = -sign(\kappa_{\chi d})$ and $|\kappa_\psi| + |\kappa_{\chi d}| < \pi$ it would mean that we have to cross the eye of the wind. If so we have to check for $u(t+T) > u_{min}$ and choose between the tack and the jibe maneuver. If the tack maneuver is chosen we would like z_1 to behave the same as if (5.30) were being used. The modulus function is given by

$$mod(a, b) = a - b\ floor(\frac{a}{b}), \tag{5.32}$$

where $floor(n)$ rounds n to the nearest integer less than or equal to n. The equation for z_1 (4.1) is

$$z_1 = \psi + \beta_b - \chi_d\cdot \tag{5.33}$$

We would like (5.33) to behave as (5.30) by using $\chi_{dt} \in [-\infty, \infty]$ instead of $\chi_d \in [-\pi, \pi]$. Inserting χ_{dt} for χ_d in (5.33), using $\beta_b \approx 0$, and setting the two equations equal to each other and solving for χ_{dt} gives

$$\begin{aligned} z_1 = \psi - \chi_{dt} &= mod(\psi - \chi_d, 2\pi) - \pi \\ \psi - \chi_{dt} &= \psi - \chi_d + \pi - 2\pi floor(\frac{\psi - \chi_d + \pi}{2\pi}) - \pi \\ \chi_{dt} &= \chi_d + 2\pi floor(\frac{\psi - \chi_d + \pi}{2\pi}) \end{aligned} \tag{5.34}$$

where χ_{dt} is the desired course angle when a tack is wanted. The jibe maneuver is then simply [2]

$$\chi_{dj} = \chi_{tack} + 2\pi sign(\kappa_{\chi d})\cdot \tag{5.35}$$

[2] given that $sign(\kappa_\psi) = -sign(\kappa_{\chi d})$ and $|\kappa_\psi| + |\kappa_{\chi d}| < \pi$ are both true.

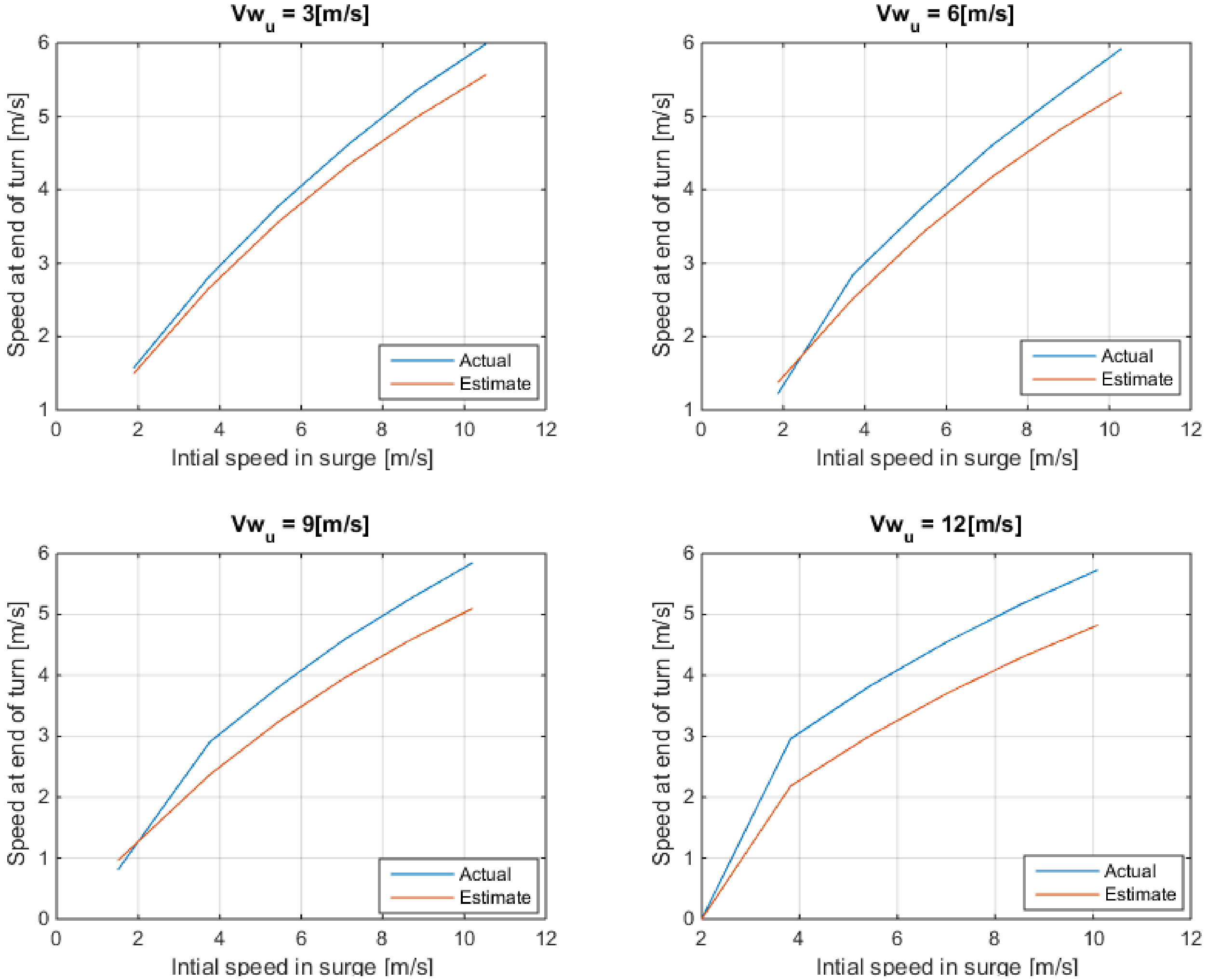


Figure 5.5: Test of surge sub-dynamic while tacking

5.4 Results

5.4.1 Surge Sub-Dynamic

By running a couple of simulations one can compare the simplified surge sub-dynamic model to the complete model. The sailboat tries to tack at different initial speeds in surge and at different wind speeds. The results are presented in Figure 5.5. The test shows that the estimates are fairly good, though the accuracy decreases when the wind speed increases. This is probably due to the sail providing some forward force to the sailboat in the beginning and at the end of the turn which our simplified model do not capture. The estimates are in general lower that the simulation values, which is a good thing considering the estimate is a lower bound

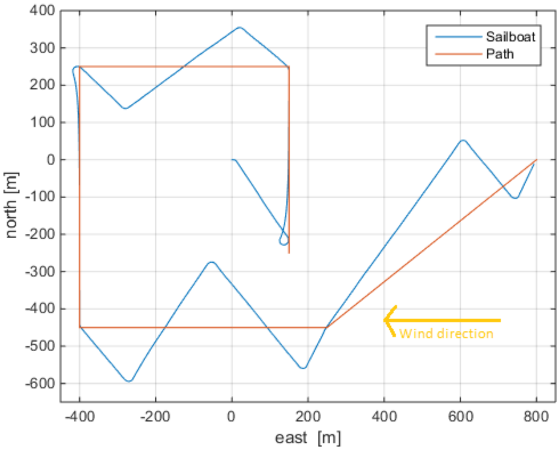


Figure 5.6: Results of path following simulation

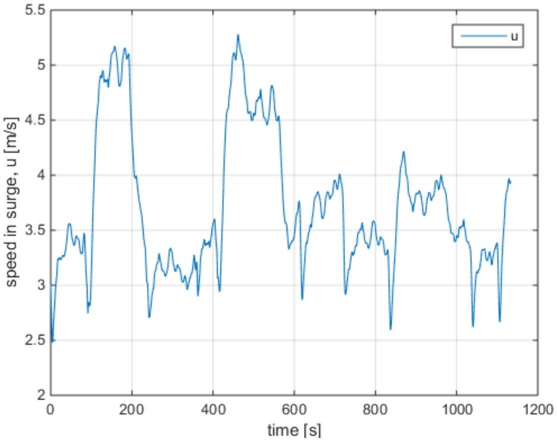


Figure 5.7: Speed in surge while path following

estimate, expect when u drops below u_{min}. This could be because the boat is not able to sustain the turning rate and thus it takes longer than anticipated to complete the turn.

5.4.2 Path Following

One simulation was executed to show how the path following algorithm works in action. The path is created such that the boat has to sail up-wind, down-wind and perpendicular to the wind. The wind speed is equal to $6\frac{m}{s}$, and the wind is coming from the east direction ($\beta_w = -90deg$). The path witch the ship tries to follow is given in Table 5.3. The initial position of the boat is $x = 0$ and $y = 0$, meaning that the sailboat starts with a large cross track error and will have to sail up-wind to get back on the path. The path restriction is $|e| < 150m$.

	p_1	p_2	p_3	p_4	p_5	p_6
$N[m]$	-250	250	250	-450	-450	0
$E[m]$	150	150	-400	-400	150	700

Table 5.3: Path used in path following test

The results are found in Figure 5.6. The boat is able to finish the path, and from Figure 5.7 one can see that the speed in surge never drops below $u_{min} = 2\frac{m}{s}$. At the beginning the sailboat manages to get back on the path, despite having to sail up-wind. It does however chose to go in the lest optimal direction in terms of minimizing the total distance traveled, which is something that might could be solved in a future iteration of this algorithm. The most important is however that it does manage to get back on the path, and when the sailboat is back on the path it does not leave the path. When the sailboat reaches the path it chooses to do a jibe maneuver, despite being the longer turning distance. The sailboat then sails along the first path segment, and when it reaches the end of the path it starts moving towards the next setpoint.

The next path segment is down wind, and the sailboat moves in the correct zig-zag pattern. When the cross track error grows larger than $150m$ the boat jibes, except in the end where it jibes a bit earlier, ensuring that sailboat hits the set point directly. Then the sailboat move southward by beam reaching, and afterwards starts beating up-wind. At the end the path moves slightly up-wind, and the sailboat is able to follow the path correctly. In general the results show that the path algorithm works as intended, and even in its simplicity its performance and robustness is quite good.

6 Sensor simulation and state estimation

6.1 Introduction

Perfect state estimation is impossible in a real system. The controller developed in chapter 4 is built upon a perfect state estimation and system knowledge, though this is impossible due to modeling errors and noisy sensors. Sensors and state estimation are a big part of a control system as it dictates which states that can effectively be used in the feedback loop and how aggressive our control action can be.

Sensor simulation will be added to the model in this chapter, and a Luenberger observer will be used for state estimation. A standard set of sensors will be implemented in the model, including GPS, gyro, accelerometer and magnetometer. An observer can, by using sensors and the knowledge about the system dynamics, improve the state estimation, reconstruct non-measured states, and provide dead reckoning capabilities in case of sensor failure. The theory presented in this chapter about state estimation and sensors are gathered from Beard and McLain [2012], Fossen [2011] and Sørensen [2013]. Wind sensor is not implemented.

Sensor bias is not included in the sensor models. This is because sensor bias is not difficult to counteract, but it would unnecessarily increase the complexity of our simulation and state estimation without giving anything insightful information regarding control.

Main contributions of chapter:

- Added realistic simulation of a standard suit of sensors.

- Implemented a Luenberger observer to estimate states of system.

6.2 Sensor simulation

6.2.1 GPS

The Global Positioning System (GPS) is used to get an estimate of the coordinates and velocity of the boat. It is a satellite-based navigation system and is a critical enabling technology for unmanned vehicles. There are 24 satellites that orbit the earth, and there are always at least four satellites observable from the earth's surface. By sending signals from the satellites and measuring the time it takes for the signals to arrive the position can be triangulated. Errors exist due to synchronization errors between the satellites clock and the receivers clock, and the variations of the speed of light as it goes though the ionosphere and troposphere. How close the satellites are to each other is also an important variable when discussing the accuracy of GPS. Furthermore, the update rate on the GPS sensor is quite low, usually in the $5Hz$ range.

Velocity

The GPS sensor is able to measure the velocity of the ground speed by using carrier phase Doppler measurements from the GPS satellite signals, and it has a standard deviation of 0.01 to 0.05 $\frac{m}{s}$. The GPS measures the speed in north and east direction, referred to as $GPS_{m,\dot{N}}$ and $GPS_{m,\dot{E}}$ respectively, from which course and ground speed can be estimated:

$$\sqrt{u^2+v^2} = \sqrt{GPS_{m,\dot{N}}^2 + GPS_{m,\dot{E}}^2} \tag{6.1}$$

$$\chi = arctan2(GPS_{m,\dot{E}}, GPS_{m,\dot{N}}), \tag{6.2}$$

where $GPS_{m,\dot{N}}$ and $GPS_{m,\dot{E}}$ are modeled as

$$\begin{bmatrix} GPS_{m,\dot{N}} \\ GPS_{m,\dot{E}} \end{bmatrix} = \sqrt{u^2+v^2} \begin{bmatrix} cos(\psi) \\ sin(\psi) \end{bmatrix} + \begin{bmatrix} \eta_{GPS,\dot{N}} \\ \eta_{GPS,\dot{E}} \end{bmatrix} \tag{6.3}$$

where $\eta_{GPS,\dot{N}}$ and $\eta_{GPS,\dot{E}}$ are zero-mean Gaussian noise with a standard deviation of $\sigma^2_{GPS,\dot{N}} = \sigma^2_{GPS,\dot{E}}$. Using an estimate of the heading angle and heeling angle, $\hat{\psi}$ and $\hat{\phi}$, one can then estimate u and v as

$$\begin{bmatrix} GPS_{m,\dot{u}} \\ GPS_{m,\dot{v}} \end{bmatrix} = J_{2D}(\hat{\phi}, \hat{\psi})^{-1} \begin{bmatrix} GPS_{m,\dot{N}} \\ GPS_{m,\dot{E}} \end{bmatrix}. \tag{6.4}$$

The $5Hz$ update frequency of the sensor is modeled using a zero-order hold function.

Position

Accurate simulation of GPS measurements for long periods of time are difficult to do because of the movement of the satellites and the dynamic of the error. However, for small periods of time, the sensor can be simulated by random noise and a bias. The bias term will not be included in this model, though attention should be made when making a real system. It will also be assumed that instead of coordinates, direct measurements of the x and y positions are obtained. The following simplified model is used:

$$\begin{bmatrix} GPS_{m,N} \\ GPS_{m,E} \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} \eta_{GPS,N} \\ \eta_{GPS,E} \end{bmatrix}, \tag{6.5}$$

where $GPS_{m,N}$ and $GPS_{m,E}$ is the sensor reading of the position in x and y, and $\eta_{GPS,N}$ and $\eta_{GPS,E}$ are random errors (not zero-mean Gaussian noise) with a magnitude of $0.4m$. The $5Hz$ update frequency of the sensor is modeled using a zero-order hold function. By combining (6.3) and (6.6) one can make an improved estimate of the position:

$$\begin{bmatrix} GPS_{m,Ni} \\ GPS_{m,E} \end{bmatrix} = \begin{bmatrix} GPS_{m,Ni} \\ GPS_{m,E} \end{bmatrix} + \frac{1}{2h} \begin{bmatrix} GPS_{m,\dot{N}} \\ GPS_{m,\dot{E}} \end{bmatrix}, \tag{6.6}$$

where h is the update frequency of the GPS. The improved estimate corrects for the low update frequency of the GPS signal. This new signal will on average be correct, compared to (6.6) which would cause a steady state error.

6.2.2 Accelerometer

The accelerometer measures the acceleration on the body in x', y' and z' axis. This includes acceleration caused by linear acceleration, coriolis acceleration and gravitational acceleration. However, in combination with a gyro, the coriolis acceleration can be removed from the measurements. To make things more convenient, the effects of the coriolis acceleration are removed from the model, removing the need for this extra step that would not add much to the accuracy of the simulation model. The accelerometer has noise, but usually little bias. The accelerometer is implemented as follows:

$$Acc_m = \begin{bmatrix} \dot{u} \\ \dot{v} \\ 0 \end{bmatrix} + \begin{bmatrix} \eta_{acc,x'} \\ \eta_{acc,y'} \\ \eta_{acc,z'} \end{bmatrix} + J_{acc}^T \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix}, \tag{6.7}$$

where

$$J_{acc} = \begin{bmatrix} cos(\psi) & -sin(\psi)cos(\phi) & sin(\psi)sin(\phi) \\ sin(\psi) & cos(\psi)cos(\phi) & -cos(\psi)sin(\phi) \\ 0 & sin(\phi) & cos(\phi) \end{bmatrix} \tag{6.8}$$

is the transformation matrix from n-frame to b-frame of x, y, z to x', y', z' and $g \frac{m}{s^2}$ is the gravitational constant and $\eta_{acc,x'}$, $\eta_{acc,y'}$ and $\eta_{acc,z'}$ are zero-mean Gaussian noise with variance σ^2_{acc}.

Assuming that the gravitational vector is dominating the measurements, i.e. $|\dot{u}| \ll g$ and $|\dot{v}| \ll g$, one can write (6.7) as

$$Acc_m = J'_{acc} \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} = \begin{bmatrix} 0 \\ sin(\phi) \\ cos(\phi) \end{bmatrix} g, \tag{6.9}$$

and it can be seen that the the heeling angle, ϕ, can be estimated by

$$Acc_{m,\phi} = arctan2(Acc_{m,y'}, Acc_{m,z'})\cdot \tag{6.10}$$

6.2.3 Gyro

Rate gyros are used to measures the rotational velocity along the different axis of the ship. It works by the principal of coriolis acceleration which is proportional to the velocity and the rate of rotation. The gyro has a significant bias which is sensitive to temperature, meaning that the sensor have to be calibrated before each use, and maybe even multiple times during long missions. The bias is not implemented in this model, though the process of recalibrating the bias while on a mission could be a point of interest. However, the sensor has some noise, and the sensor measurements is implemented as

$$Gyro_m = \begin{bmatrix} Gyro_{m,p} \\ Gyro_{m,r} \end{bmatrix} = \begin{bmatrix} p \\ r \end{bmatrix} + \begin{bmatrix} \eta_{gyro,p} \\ \eta_{gyro,r} \end{bmatrix}, \tag{6.11}$$

where $Gyro_m$ is the sensor reading and $\eta_{gyro,p}$ and $\eta_{gyro,r}$ are zero-mean Gaussian noise with variance σ^2_{gyro}.

6.2.4 Magnetometer

The magnetometer measures the earth's magnetic field. The earth's magnetic field is three dimensional with north, east and down components that vary with the location on earth. A compass provides an indication of the heading relative to the magnetic north, but do not work well for big heel and pitch angles. Other sources for noise are nearby metal or electrical interference. Modern digital compasses uses three-axis magnetometers, which can be used to compensate measurements at big heel and pitch angles by effectively projecting the three dimensional vector onto the plane. To simplify our model, a two-axis magnetometers is used, and it will always be level (no heel or pitch angle) and the magnetic north is equal to the true north. A more complex model could be implemented though the extra detail would not contribute a lot to the realism of the sensor simulation. The simplified model is

$$Mag_m = \psi + \eta_{mag}, \tag{6.12}$$

where Mag_m is the sensor readings and η_{mag} is zero-mean Gaussian noise with variance σ^2_{mag}.

6.3 Observer

A disadvantage associated with filtering is phase lag. This reduces the performance of the controller as the states in the feedback loop effectively has a delay. An observer is able to overcome this issue by the use of a prediction element in the state estimation. This prediction is only possible when the system dynamics are know or can be approximated. In this thesis, a simple continues Luenberger observer is used, which is described as

$$\begin{aligned}\dot{\hat{x}} &= A\hat{x} + Bu + L(y - \hat{y}) \\ \hat{y} &= C\hat{x} + Du\end{aligned}, \tag{6.13}$$

where $\hat{x} = [\eta \ \nu]^T$ is the estimation of states, L is the gain of the observer, y is the states that can be measured using sensors and $\hat{y}$ is the estimation of those states. It can be shown that the error, $e = x - \hat{x}$, has a dynamic of

$$\dot{e} = (A - LC)e, \tag{6.14}$$

which goes asymptotically to zero if $A - LC$ is Hurwitz (all eigenvalues are negative). The dynamics of the sailboat are non-linear, and to make matters worse, can not be linearized easily. The Luenberger observer can however be modified for a non-linear system:

$$\begin{aligned}\dot{\hat{x}} &= f(\hat{x}) + B(u) + L(y - \hat{y}) \\ \hat{y} &= C\hat{x} + D(u)\end{aligned}, \tag{6.15}$$

where $f(\hat{x})$ represents the system dynamics. The system model developed in chapter 3 includes the forces from the actuators, i.e. $B(u) = 0$, and it is easy to see that $D(u) = 0$. The Luenberger observer (6.15) is then simplified to

$$\begin{aligned}\dot{\hat{x}} &= f(\hat{x}) + L(y - \hat{y}) \\ \hat{y} &= C\hat{x}\end{aligned}. \tag{6.16}$$

Using the system model (3.2), $f(\hat{x})$ can be formulated as

$$f(x) = \begin{bmatrix} J(\phi,\psi)\nu \\ M^{-1}(-C_{RB}(\nu)\nu - C_A(\nu_r)\nu_r - D(\nu_r) - g(\eta) + S + K + R) \end{bmatrix}, \tag{6.17}$$

where $M = M_{RB} + M_A$. The measured states, y:

$$y = \begin{bmatrix} \eta_m \\ \nu_m \end{bmatrix}, \tag{6.18}$$

where

$$\eta_m = \begin{bmatrix} GPS_{m,Ni} \\ GPS_{m,Ei} \\ Acc_{m,\phi} \\ Mag_m \end{bmatrix} \quad (6.19) \qquad \nu_m = \begin{bmatrix} GPS_{m,u}(\hat{\phi},\hat{\psi}) \\ GPS_{m,v}(\hat{\phi},\hat{\psi}) \\ Gyro_{m,p} \\ Gyro_{m,r} \end{bmatrix}. \quad (6.20)$$

C in equation (6.16) then simply becomes the identity matrix of size eight. For linear systems the observer gain, L, can be found using the Kalman gain. The Kalman gain is optimized for systems where the noise in both the system dynamics and measurements can be modeled as zero-mean Gaussian noise. When the variance of each sensor and dynamic is know the Kalman gain is the optimal gain. For non-linear systems, the extended kalman gain can be used, which is just the regular kalman gain calculated based on a linearization of the system dynamics at each time step. The variance of each sensor is easy to find, but the same is not true for the system dynamics. The L gain was thus just tuned manually until satisfactory performance was achieved.

If V_w was know perfectly the model of our system would be perfect, and it follows that L would be zero. Instead, only the average wind speed is known, $\hat{V_w}$, which effectively makes the wind gusts, and the variable wind direction, the noise in the system. The sail is affected by the wind speed squared, and it follows that the size of the noise in the system is related to the square of the wind speed. A time-varying observer gain based on the size of the wind speed could be one possibility, though this is too complicated. Instead, the observer gain is tuned for $U_{10m} = 10\frac{m}{s}$, a fairly high wind speed which can be seen as a worst case scenario. Further more, depending on if the boat is running or is beam reaching, the optimal observer gain might be different as the noise propagates through the system differently. This has to be taken into consideration when tuning manually. The observer gain can be found in appendix B.

6.4 Results

6.4.1 Observer

One simulation was executed, showing the estimation of states while running, as well as the sensor simulation. The average wind speed and direction was known to the observer, and and true wind speed used in the simulation is $6\frac{m}{s}$. The results show the last 5 seconds of a 50 second simulation.

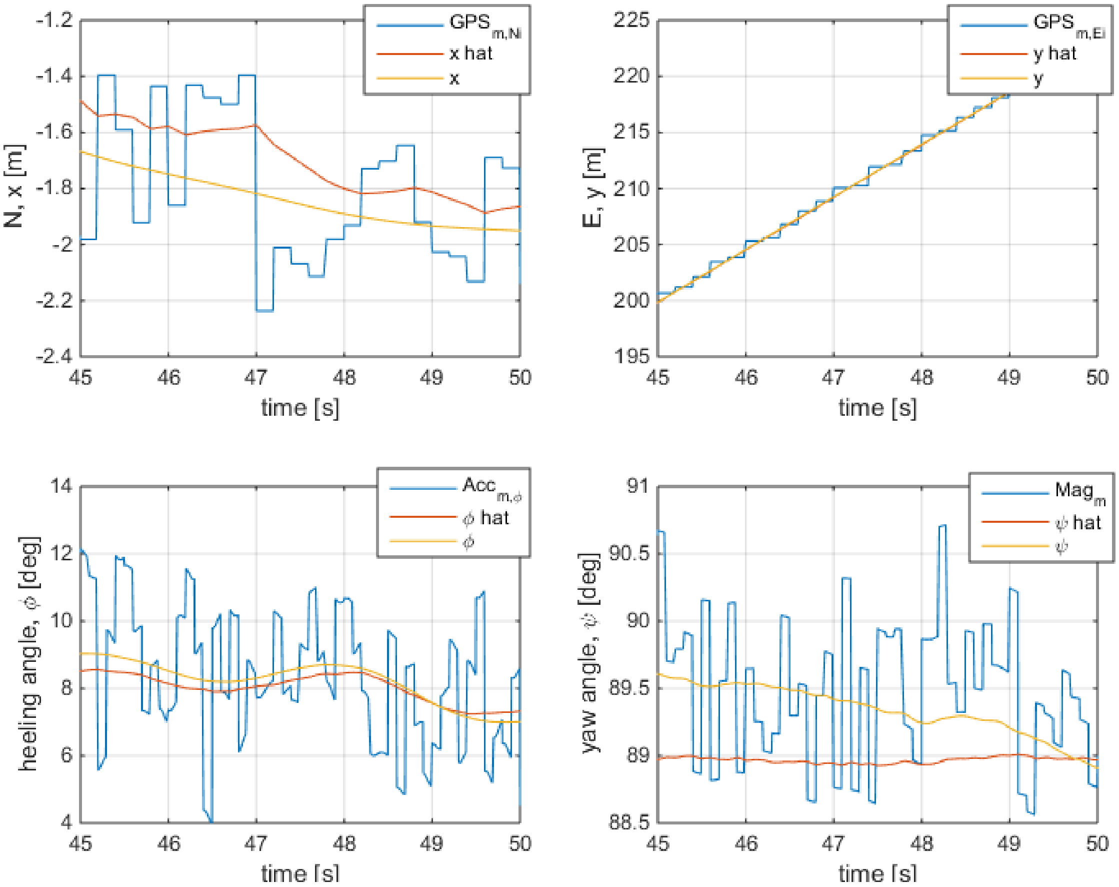


Figure 6.1: State estimation, η

Figure 6.1 shows the sensor reading, estimation and real value of the states in η. In the first plot one can see that the GPS position simulation works as intended; the error looks random between $\pm 0.4m$, though on average the estimate seems correct. The observer estimate of x looks correct as well, though for some control schemes the estimate might not be steady enough. The next plot shows the estimate and sensor reading of state y, and the results is similar to that of x.

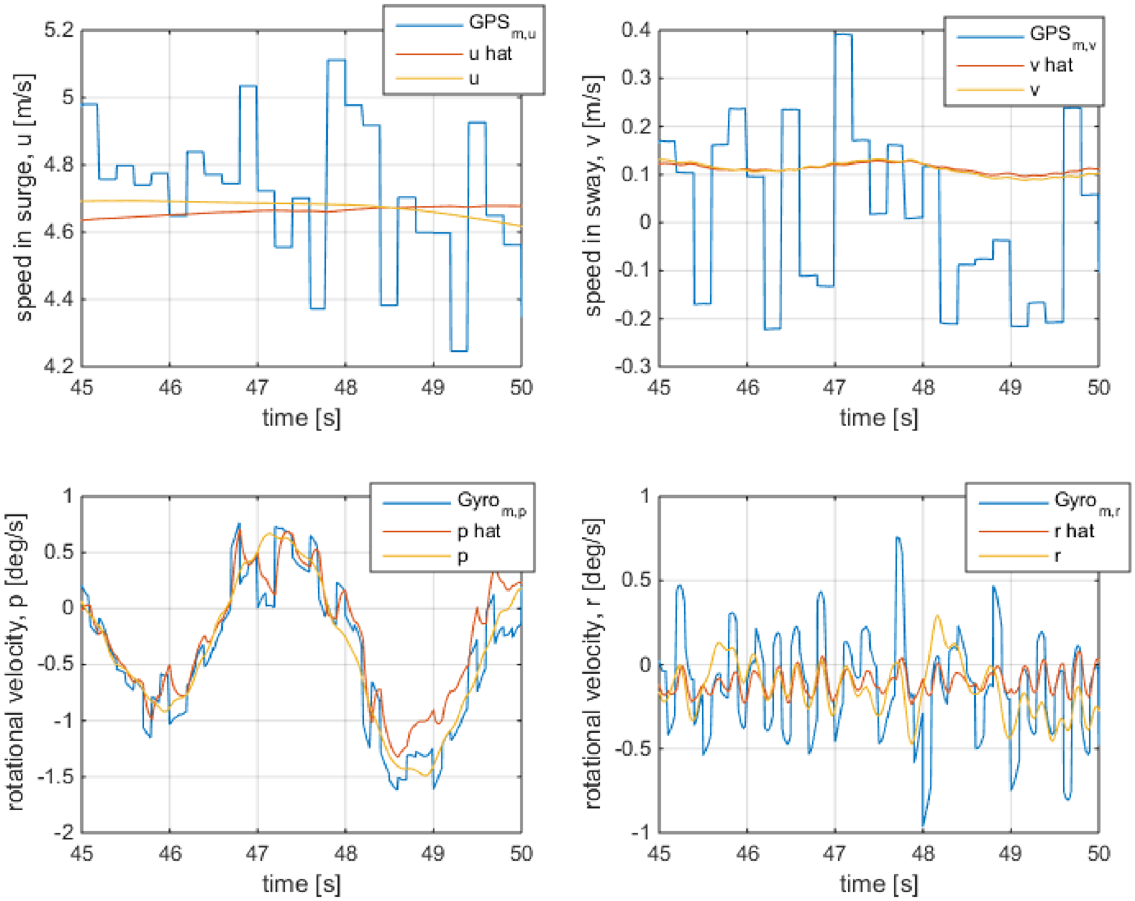


Figure 6.2: State estimation, ν

The next two plots shows the result for ϕ and ψ. The estimation of heeling angle by the use of accelerometer is working, even with an substantial amount of noise. Using the accelerometer measurements alone would probably not give the same results, and the estimate of ϕ is strongly connected to the estimate of p, which is rather good (Figure 6.2) due to the accuracy of the gyro. The estimate of ψ could probably be better, in terms of average error, by increasing the L gain. However, this would make $\hat{\psi}$ less steady which could lead to an unnecessary amount of control action in the rudder.

Figure 6.2 shows the results for the states in ν. The first two plots shows u and v. The estimation of u and v are very close to the real value, and the simulation of the GPS position sensor is good. The estimates of p and r are very accurate as well, and the gyro simulations looks reasonable.

7 Simple Controller

7.1 Introduction

The controller in chapter 4 could be considered ideal, but it assumes perfect state estimation for its feedback-loop. Further more, the course controller tries to cancel all nonlinearities in the system, but this requires a very precise model of the system which can be difficult, or even impossible, to make. The ideal controller also takes into account a lot of non-important details and dynamics, which makes the controller unnecessarily complex.

In this chapter a simpler controller will be developed based on the results from the previous chapters. The main objectives of the simple controller is to provide a simpler to implement option, while minimizing the impact on performance. This is achieved by analyzing the results from chapter 3 and looking at what forces and moments that can be neglected. The ideal controller from chapter 4 will be the base of the controller developed in this chapter, though some elements will be changed, and other parts removed.

As in chapter 4, there will be two main independent controllers. One to control the sail, and another one that controls the yaw and course. The biggest changes in the controller will relate to the course controller and roll reduction controller, as thees are the most dependent on the system dynamics and estimated states in their feedback loops.

Main contributions of chapter:

- A simple to implement course controller with high performance.
- A simple to implement roll controller with high performance.

7.2 Rudder Controller

7.2.1 Course Control

The desired momentum of the rudder was found in chapter 4 to be equal to (4.20)

$$\begin{aligned} u = &- C_\psi(S + K - C_A(\nu_r)\nu_r - D(\nu_r)) \\ &- (I_{zz} - N_{\dot r})(-\ddot{\chi}_d + K_P(z_2 - K_P z_1) + z_1 + K_D z_2)\cdot \end{aligned} \tag{7.1}$$

Looking at the force analysis from chapter 3, Figure 3.14 and 3.17, reveals that many of thees moments contribute little to the yaw sub-dynamic. When running, it is only the sail that has a significant moment contribution in yaw, though the dynamics while beam reaching are more complex. While beam reaching, the keel, sail and added mass Coriolis are all important. Assuming $r \approx 0$, the added mass Coriolis term (3.9) in yaw can be simplified to

$$C_\psi C_A(\nu_r)\nu_r \approx -(Y_{\dot v} - X_{\dot u})u_r v_r \approx -\frac{1}{2}(Y_{\dot v} - X_{\dot u})U^2 sin(2\beta_b), \tag{7.2}$$

where $U^2 = u^2 + v^2$. This is simply the Munk moment.

Assuming that the correction term, β_b, is correct, the force created by the keel should be equal to F_U (4.31). From (3.29) and $\beta_{ck} \approx \pi$ it follows that the moment created in yaw by the keel should be equal to

$$K_\psi = -F_U l_3\cdot \tag{7.3}$$

Figure 3.14 and 3.17 only shows the moments in yaw while in a steady course, and the dampening term is thus very low. When turning there will be a lot more dampening in yaw, but it is considered a "good" nonlinearity because it increases the stability of the system (Fossen [2011], p. 457). It can be proven that the dampening term dissipates energy from the second Lyapunov function. The equation for z_2 can be simplified when $z_1 \approx 0$ and $\dot{\chi}_d \approx 0$ to

$$z_2 = r - \alpha = r - \dot{\chi}_d + K_P z_1 \approx r\cdot \tag{7.4}$$

When v is small the dampening in the yaw sub-dynamic can be approximated by

$$D_\psi = d_{l_{66}} r + D_{q_N}(\nu_r) \approx d_{l_{66}} r + D_C r|r|, \tag{7.5}$$

where $D_C > 0$. If u do not cancel out the dampening term, D_ψ, in the ideal controller (7.1) then the second Lyapunov function becomes

$$\dot{V}_2 = -K_P z_1^2 - K_D z_2^2 - z_2\Big(\frac{d_{l_{66}} r + D_C r|r|}{I_{zz} - N_{\dot r}}\Big)\cdot \tag{7.6}$$

By inserting the simplified expression for z_2 (7.4) in the last term it is easy to see that $\dot{V}_2 < 0$:

$$\dot{V}_2 = -K_P z_1^2 - K_D z_2^2 - r^2\Big(\frac{d_{l66} + D_C|r|}{I_{zz} - N_{\dot{r}}}\Big). \tag{7.7}$$

Removing the moments that do not contribute to the yaw sub-dynamic, inserting the simplifications of the added mass coriolis term (7.2) and keel (7.3), removing the dampening term and adding integral control action to account for any errors (in estimation, modeling or by simplification), the control input u is given by

$$\begin{aligned} u = &- F_U l_3 - S_\psi - \frac{1}{2}(Y_{\dot{v}} - X_{\dot{u}})U^2 sin(2\beta_b) \\ &+ (I_{zz} - N_{\dot{r}})(-\ddot{\chi}_d + K_p(z_2 - K_p z_1) + z_1 + K_D z_2) + K_{ir}\int_0^t z_1 \tau. \end{aligned} \tag{7.8}$$

Looking at Figure 3.17, K_ψ, S_ψ and the Munk moment are all slowly changing moments. A very simple controller could thus replace these three terms with integral control action, though it would reduce performance after changing course as the integral term would need time to stabilize again, which is done frequently when sailing up-wind and down-wind.

7.2.2 Correction Term, β_b

The undesirable force, F_U (4.31), can also be simplified. Assuming a low drift angle one can simplify this to just the forces in negative y-direction, as $cos(-\frac{\pi}{2}) \approx 0$ and $sin(-\frac{\pi}{2}) \approx -1$. When the drift angle is low it follows that v is small, which implies $D_y(\nu_r) \approx 0$. Furthermore, $R_{y'}$ can be approximated by $R_{y'} \approx -\frac{u}{l_4}$, where u is the desired yaw moment of the rudder. This gives the following simplified solution for F_U:

$$F_U \approx -S_{y'} + \frac{u}{l_4}. \tag{7.9}$$

In (3.28), V_{ck} is used in the calculation of α_k, but by assuming $p = r \approx 0$ it can be simplified to

$$\alpha_k = \frac{2F_U}{p_w A_k C_{L0_k} U^2}. \tag{7.10}$$

7.3 Sail Controller

7.3.1 Roll Controller

The roll controller developed in chapter 4 removed all the nonlinearities in the system and then used a linear quadratic regulator (LQR) to control the roll motion. A simplified controller can be made by linearizing the system in roll about an equilibrium point, which is selected to be the desired heel angle. The linearized system can be written as

$$x = \begin{bmatrix} \phi \\ p \end{bmatrix} \qquad (7.11) \qquad\qquad \dot{x} = Ax + Bu\cdot \qquad (7.12)$$

Looking at the results of the force analysis in chapter 3, Figure 3.14 and 3.17 shows that the restoring moment and the sail contributes roughly 91% of the total moment generated in roll, and the keel another 8%. Assuming F_U is correct, then from (3.29), the moment created by the keel is equal to $-F_U h_2$. The restoring moment (3.10) can be linearized to be written as

$$G_\phi = p_w g \Delta GM_t((1 - 2sin(\phi_w)^2)\phi + C_1, \tag{7.13}$$

where ϕ_w is the equilibrium point and C_1 is equal to

$$C_1 = p_w g \Delta GM_t cos(\phi_w) sin(\phi_w)\cdot \tag{7.14}$$

Matrices A and B in our linearized system can then be written as

$$A = \begin{bmatrix} 0 & 1 \\ -\frac{p_w g \Delta GM_t(1-2sin(\phi_w)^2)}{I_{xx}-K_{\dot{p}}} & 0 \end{bmatrix}, \tag{7.15}$$

$$B = \begin{bmatrix} 0 \\ \frac{1}{I_{xx}-K_{\dot{p}}} \end{bmatrix}, \tag{7.16}$$

where u is defined as

$$u = -K_{LQR} \begin{bmatrix} \phi - \phi_d \\ p \end{bmatrix} - K_{is} \int_0^t \phi - \phi_d d\tau + C_1 + F_U h_2, \tag{7.17}$$

and

$$S_{\phi r_d} = \frac{u}{V_{ws}^2}\cdot \tag{7.18}$$

The LQR controller gain, K_{LQR}, was then tuned by using the same matrices Q and R as in chapter 4., which gave the result $K_{LQR} = [9.03e4\ 1.04e5]$. In a real system the constant C_1 can be estimated by the use of the integral action, as the steady sate value of the integral action will approximately be C_1. Furthermore, $F_U h_2$ is fairly small, and by the force analysis in chapter 3, produced a steady moment. It follows that the integral action could handle the error caused by not including it. However, if F_U is already calculated, there is little reason not to include it. As already discussed, it would however reduce performance after changing course, which is done frequently when sailing up-wind and down-wind.

7.4 Results

Three different simulation setups are used in the following simulations. In the first simulation setup, the ideal controller is used along with perfect state estimation. In the second, the simple controller is used along perfect state estimation. In the last, the simple controller is used, but the observer is used to estimate states and only the average wind speed is known.

7.4.1 Course controller

The first scenario is of the boat trying to keep a steady course angle of 0 degrees (north) when the mean wind speed is $5\frac{m}{s}$ and the angle of the wind is equal to $\beta_w = 125deg$. The simulations has a duration of 1000 seconds. Thees are the same conditions that were used when testing the course controller in chapter 4, except that the simulation time has been increased.

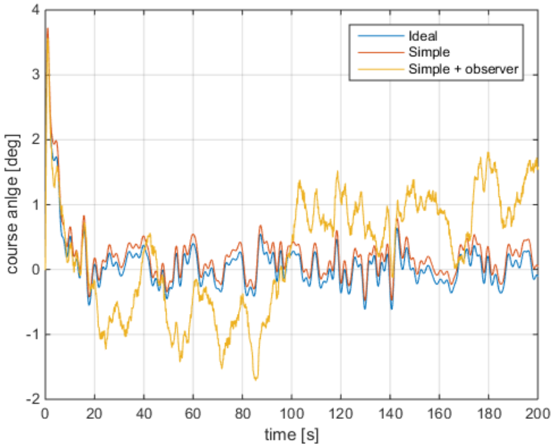


Figure 7.1: Course controller simple, χ-plot

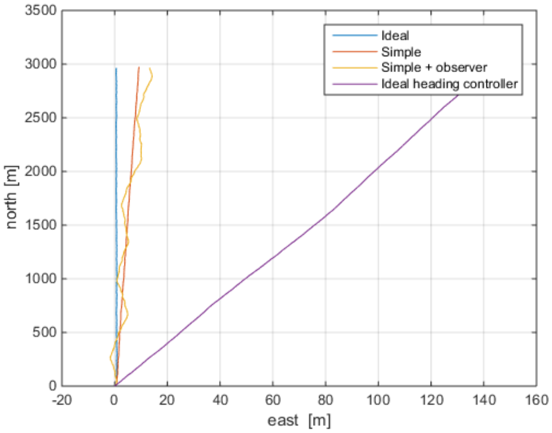


Figure 7.2: Course controller simple, NE-plot

The overall drift from the desired course can be estimated as it where in chapter

4. To compute an estimate of the error, $|atan(\frac{x}{y})|$ is used, where x is the distance traveled in north direction and y is the distance traveled in east direction. The course controller takes 20 seconds to adjust, and the drift caused by this is removed from the error. See table 7.1 for results.

Table 7.1: Results of course controller test

scenario	x $[m]$	y $[m]$	error $[deg]$
(ideal) heading controller	149	3100	2.71
ideal	0	2950	0
simple	9	2950	0.17
simple + observer	10	2950	0.19

The simple controller with perfect state estimation perform slightly worse than the ideal controller, and the simplifications done to the drift correction term causes roughly a $0.17deg$ error. Otherwise the controller performs quite similarly to the ideal controller, which is easy to see on in Figure 7.2. The drift is slightly larger when the simple controller has to use the observer and average wind estimate, and the course fluctuates considerably more. In the ideal controller showed that without the drift correction term the error would be $2.71deg$, and it follows that the drift from the course has been reduced by roughly 1396% compared to the ideal controller without drift correction.

The variation in course angle is considerable less when perfect state estimation is used, and the effectiveness of the controller is mostly limited by how well one can estimate the states used in the feedback loop, and in that regard the simplified controller performed excellent. However, perfect state estimation or not, the controller is able to keep the course without too much deviation. The amplitude of the fluctuating seems to be around 1m, which is not bad at all.

The next scenario where designed to test performance while turning. A lot of the simplifications of the ideal controller were justified on the basis of holding a steady course, such as $r \approx 0$ when simplifying the Coriolis added mass term. To make the result as focused on the controller as possible, $V_w = [0\ 0]^T$, $\dot{\nu_u}$ and perfect state estimation were used. The boat will try to do a $90deg$ turn at $u = 5\frac{m}{s}$, no wind.

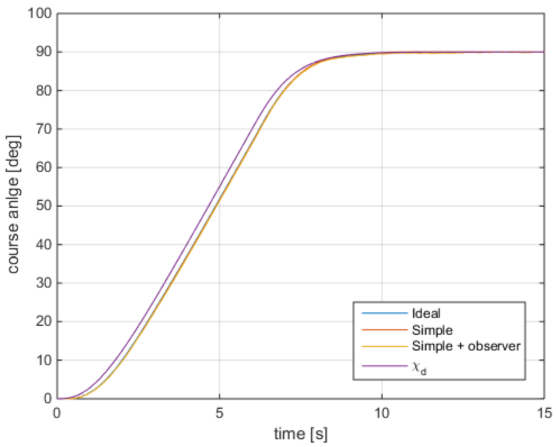


Figure 7.3: Course controller simple, time evolution of χ

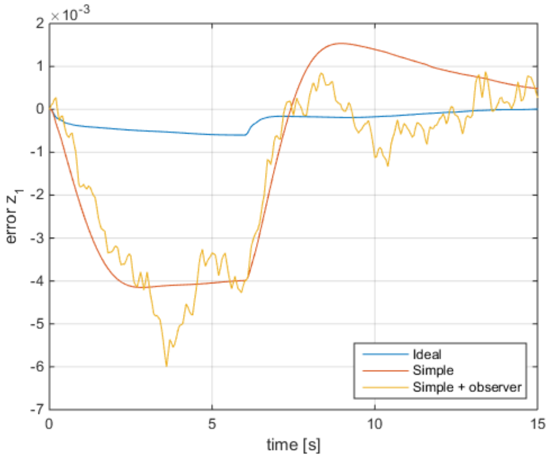


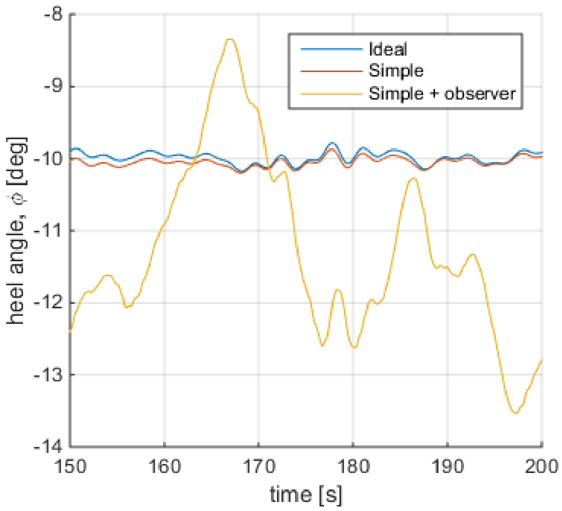
Figure 7.4: Course controller simple, time evolution of z_1

From Figure 7.3 it is difficult to see the difference in performance between the three controller setups, but all of them works as they should. There are no overshoot and the turning rates are stable in all simulations. Figure 7.4 shows the time evolution of z_1, and it is easier to see the difference between the two controllers in this Figure. The simple controller has a larger constant error in z_1 compared to the ideal controller, though this is to be expected because the drag term has been neglected. When state estimation is turned on the error in z_1 fluctuates a bit due to noise in the estimates by the observer, but it does not seem to effect performance or overall stability.

7.4.2 Roll controller

The last set of simulations shows how the different setups handles reducing roll motion. The boat is set to go on a course direction perpendicular to a mean wind speed of $10\frac{m}{s}$. The results will show the last 50 seconds of a 200 second simulation, ensuring that the top speed in surge has been reached and that the integral term in the controller has had time to reach a steady value. These are the same conditions that were used when testing the roll controller in chapter 4.

Figure 7.5 and 7.6 shows that the simple controller, even when using the observer for state estimation, is able to reduce the heeling angle and roll motion. The simple controller performs almost identical to the ideal controller, justifying the simplifications. When the true value of the states are not known we get a big drop in performance. The drop in performance is most likely caused by that only the average wind speed is known, which makes it difficult to calculate the relative desired momentum in roll ($S_{\phi r_d}$). However, the heeling angle is kept around $10deg$ regardless of which controller setup is used, which is the main purpose of the roll controller.



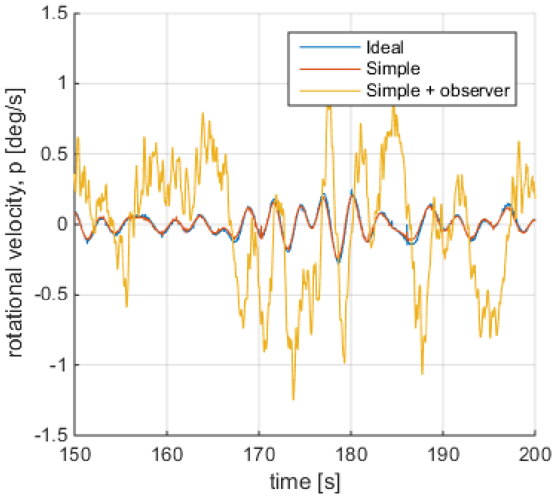


Figure 7.5: Roll controller simple, time evolution of ϕ

Figure 7.6: Roll controller simple, time evolution of p

In comparison, without a roll controller the heeling angle was about $25deg$.

In Table 7.2 the results are summarized. Besides reducing the heeling angle to a safe level, the simple controller without perfect state estimation is able to reduce the amplitude of the roll motion by about 47.8% and the rotational velocity by 62.5% compared to having no controller in roll at all.

Table 7.2: Results of roll controller test

scenario	amplitude of ϕ $[deg]$	maximum p $[\frac{deg}{s}]$	u $[\frac{m}{s}]$
ideal	0.24	0.3	6.48
simple	0.24	0.3	6.52
simple + observer	2.54	1.5	6.54
controller off	5.31	4.0	8.3

7.4.3 Path following

The following simulation is the same as in chapter 5, except that the simple controller is used, and only the average wind estimate is known.

From Figure 7.7 and 7.8 it is easy to see that the sailboat manages to follow that path. The result is very similar to the simulation in chapter 5 where the ideal controller with perfect state estimation were used. The simple controller is able to tack and jibe, sail close hauled, beam reaching and broad reaching, and the speed in surge

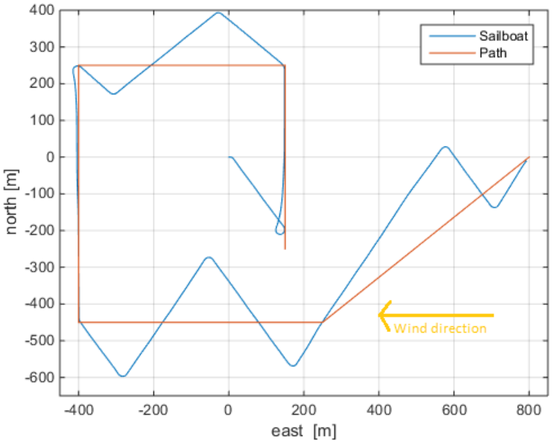


Figure 7.7: Results of path following simulation

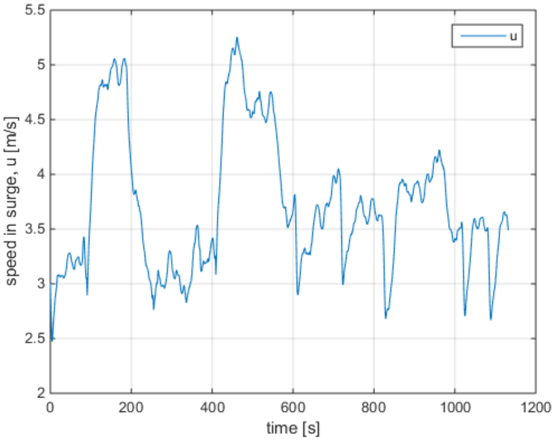
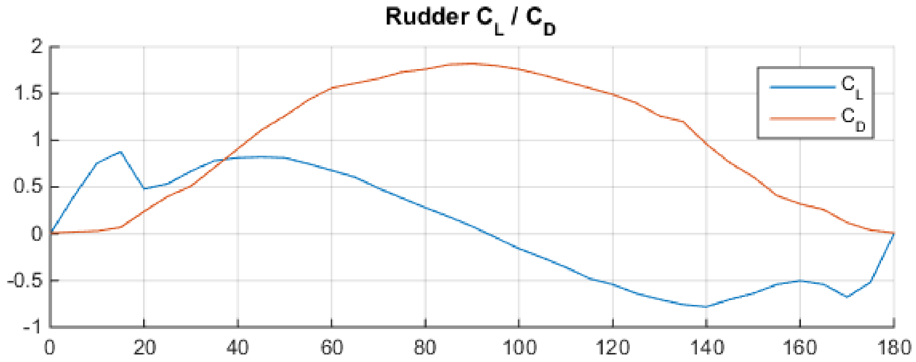


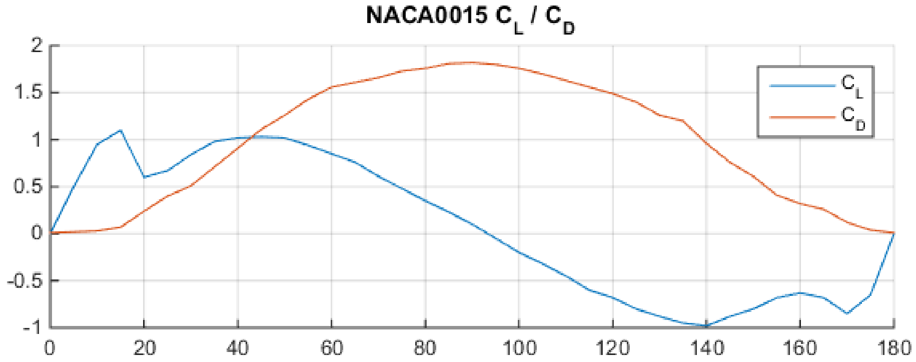
Figure 7.8: Speed in surge while path following

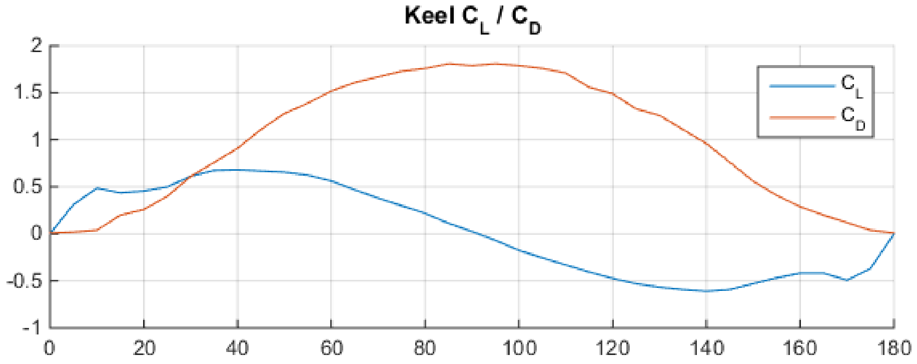
never drops below u_{min}. The speed in surge is comparable to the ideal controller, and it manages to complete the path in a similar amount of time.

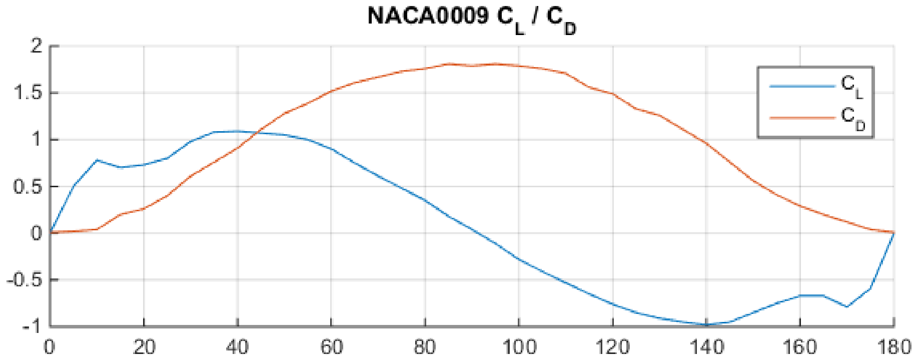
8 Appendices

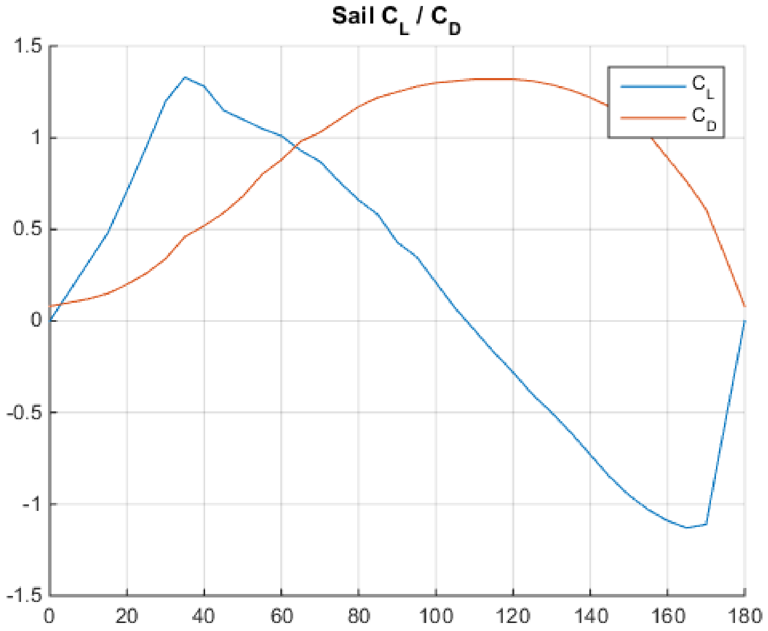
A

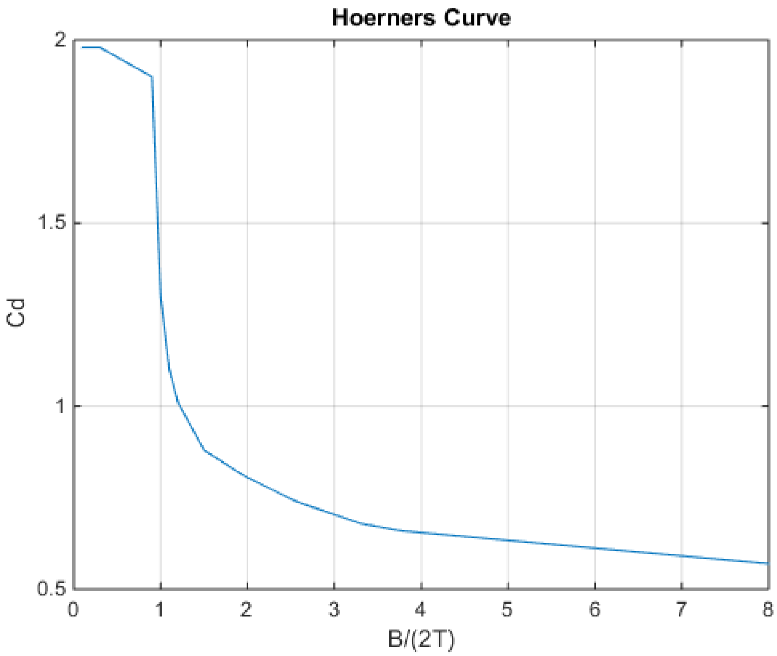





Keel C_L / C_D
C_L
C_D
2
1.5
1
0.5
0
-0.5
-1
0
20
40
60
80
100
120
140
160
180


NACA0009 C_L / C_D
C_L
C_D
2
1.5
1
0.5
0
-0.5
-1
0
20
40
60
80
100
120
140
160
180

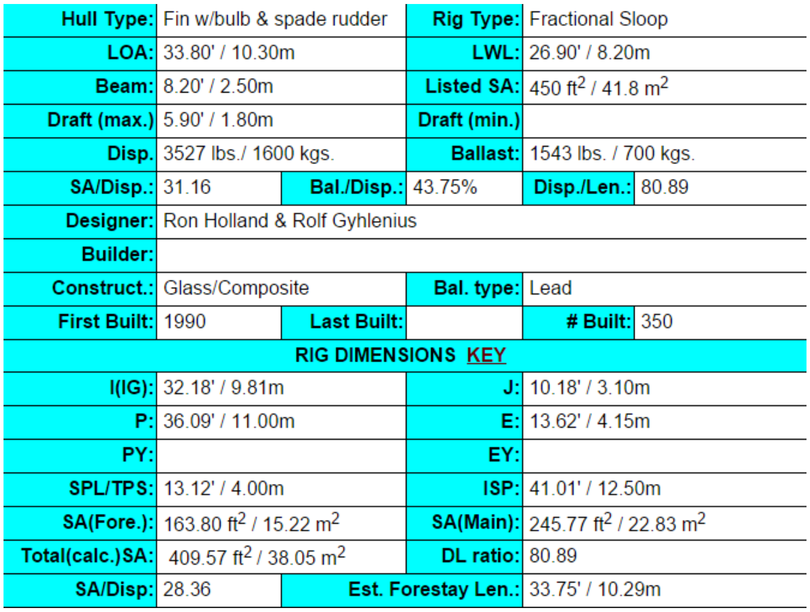

Sail C_L / C_D
C_L
C_D
1.5
1
0.5
0
-0.5
-1
-1.5
0
20
40
60
80
100
120
140
160
180


Hoerners Curve
Cd
B/(2T)
2
1.5
1
0.5
0
1
2
3
4
5
6
7
8

B

Data Sheet and Parameters used in Simulation

Data sheet and drawings of sailboat were found on the website sailboatdata.com (`http://sailboatdata.com/viewrecord.asp?class_id=2893`), accessed on 9-10-2015. All estimated parameters are calculated in the script called "CalcSailBoatValues.m".



Hull Type:	Fin w/bulb & spade rudder		**Rig Type:**	Fractional Sloop	
LOA:	33.80' / 10.30m		**LWL:**	26.90' / 8.20m	
Beam:	8.20' / 2.50m		**Listed SA:**	450 ft^2 / 41.8 m^2	
Draft (max.)	5.90' / 1.80m		**Draft (min.)**		
Disp.	3527 lbs./ 1600 kgs.		**Ballast:**	1543 lbs. / 700 kgs.	
SA/Disp.:	31.16	**Bal./Disp.:**	43.75%	**Disp./Len.:**	80.89
Designer:	Ron Holland & Rolf Gyhlenius				
Builder:					
Construct.:	Glass/Composite		**Bal. type:**	Lead	
First Built:	1990	**Last Built:**		**# Built:**	350
RIG DIMENSIONS KEY					
I(IG):	32.18' / 9.81m		**J:**	10.18' / 3.10m	
P:	36.09' / 11.00m		**E:**	13.62' / 4.15m	
PY:			**EY:**		
SPL/TPS:	13.12' / 4.00m		**ISP:**	41.01' / 12.50m	
SA(Fore.):	163.80 ft^2 / 15.22 m^2		**SA(Main):**	245.77 ft^2 / 22.83 m^2	
Total(calc.)SA:	409.57 ft^2 / 38.05 m^2		**DL ratio:**	80.89	
SA/Disp:	28.36	**Est. Forestay Len.:**		33.75' / 10.29m	

Figure 8.1: Datasheet

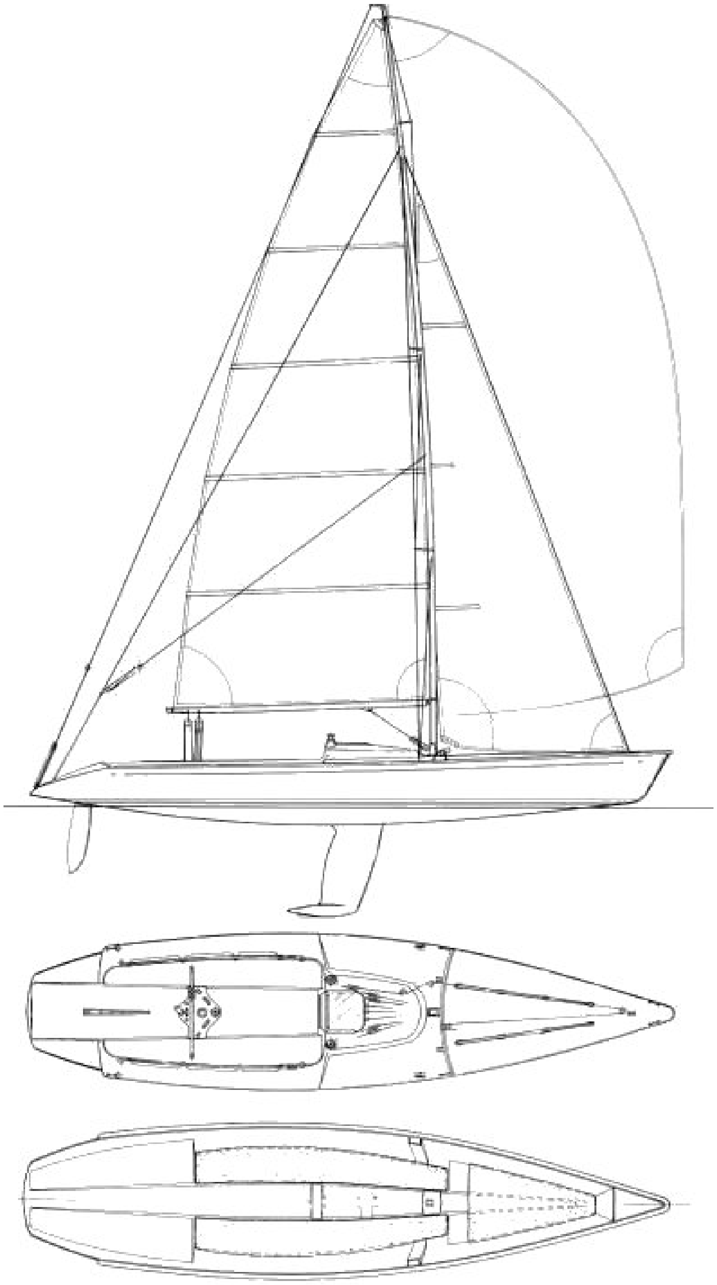


Figure 8.2: Drawing of sailboat

Table 8.1: Important lengths

Variable	**value**
h1	5.2 $[m]$
h2	0.95 $[m]$
h3	0.7 $[m]$
l1	-1.82 $[m]$
l2	1.35 $[m]$
l3	-0.66 $[m]$
l4	3.66 $[m]$
l5	2.2 $[m]$
l6	2 $[m]$

Table 8.2: Rigid and added mass

Variable	**value**
m	$1.6e^3$ $[kg]$
I_{xx}	$3.3e^3$ $[kg\ m^2]$
I_{zz}	$4.6e^3$ $[kg\ m^2]$
$X_{\dot{u}}$	-1.6e2 $[kg]$
$Y_{\dot{v}}$	-1.2e3 $[kg]$
$K_{\dot{p}}$	-1.0e3 $[kg\ m^2]$
$N_{\dot{r}}$	-2.4e3 $[kg\ m^2]$
$Y_{\dot{r}}$	-3.5e2 $[kg\ m]$

Table 8.3: Dampening

Variable	**value**
k	0.15 $[-]$
$d_{l_{11}}$	10 $[\frac{kg\ m}{s}]$
$d_{l_{22}}$	16 $[\frac{kg\ m}{s}]$
$d_{l_{66}}$	40 $[\frac{kg\ m}{s}]$
B_{F0}	15 $[-]$
B_L	70 $[\frac{kg}{rad}]$

Table 8.4: Sailboat basics

Variable	**value**
L_{WL}	8.8 $[m]$
S_H	$9.8e^4$ $[m^2]$
ω_{roll_0}	2.39 $[\frac{1}{s}]$
GM_t	2.4 $[m]$

Table 8.5: Actuators

Variable	**value**
T_r	0.2 $[s]$
T_b	0.5 $[s]$
σ_{sat}	35 $[deg]$
λ_{sat}	110 $[deg]$
Q_k	200000 $[\frac{kg}{s^2}]$
Q_b	1000 $[\frac{kg}{s}]]$

Table 8.6: Foils

Variable	**value**
Asp_k	1.75 $[-]$
Asp_r	$9.8e^4$ $[-]$
C_{L0_k}	3.57 $[-]$
C_{L0_r}	4.51 $[-]$
Keel Area	0.93 $[m^2]$
Rudder Area	0.30 $[m^2]$
Sail Area	22.8 $[m]$

Table 8.7: Controller tuning parameters

Variable	**value**
LQR_{ideal}	$[8.66e4\ 1.05e5]\ [-]$
LQR_{simple}	$0.5\ [-]$
K_{is}	$0.2\ [-]$
K_P	$0.25\ [-]$
K_D	$50\ [-]$
K_{ir}	$500\ [-]$
K_{c_+}	$200\ [-]$
K_{c_-}	$100\ [-]$
K_{ib}	$0.2\ [-]$
T_ψ	$0.66\ s$

Table 8.8: Path following

Variable	**value**
r_{sat}	$15\ [\frac{deg}{s}]$
Θ_{min}	$55\ [deg]$
Θ_{MIN}	$40\ [deg]$
Θ_{max}	$140\ [deg]$
ϵ_Θ	$5\ [deg]$
D_γ	$18\ [kg\ m]$
D_ρ	$44.7\ [kg\ m]$
$D_\S$	$1.11\ [\frac{kg}{s}]$

Table 8.9: Sensor simulation

Variable	**value**
$\eta_{GPS,\dot{N}/\dot{N}}$	$0.05\ [-]$
$\eta_{GPS,N/E}$	$0.4\ [-]$
η_{acc}	$0.1\ [-]$
η_{Gyro}	$0.00002\ [-]$
η_{Mag}	$0.0002\ [-]$

$$\text{(observer gain)}L = \begin{bmatrix} 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.05 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 20 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 5 \end{bmatrix} \tag{8.1}$$

C

Implementation and use of Simulator

Everything has been implemented in simulink (matlab). The implementation was made using matlab R2014b, and if any problems are encountered it is recommend to use this version of matlab. Comments are used in all the matlab scripts with references to this thesis. There are automated scripts to recreate any of the results presented in this paper, these are located in the folder called "plot_scripts". Simply run the desired script and all simulations and graphs are made automatically. To make it simpler to initiate a simulation there is a script called "runSimulation.m". Description of all the inputs and outputs are described in the Table below.

Table 8.10: runSimulation.m

Variable	**value**
model_name	name of simulink model
simTime	simulation time [s]
nu_init	initial value of nu on the format ’[u;v;p;r]’ [m;m;rad;rad]
eta_init	initial value of eta on the format ’[x;y;phi;psi]’ [m/s; m/s; rad/s; rad/s]
windSpeedMean	average wind speed at z = -10m [m/s]
windDirection	average wind direction in [rad]
desiredCourse	desired course if path_model>0 [rad]
desiredCourse_init	initial desired course [rad]
driftGain	should be equal to 1 to enable the drift correction term
path_model	>0 follows desired course, <0 enables path following
roll_target_init	desired max heeling angle
lamda_init	initial value of lambda (corrects l_{70} automatically) [rad]
windEstimation_mode	=0 perfect wind estimation, =2 average wind is known
observer_on	=0 perfect state estimation, =1 enables observer
controller_mode	=0 ideal controller, =1 simple controller

Before simulations can be initiated there are a couple of settings that must be adjusted. Global variables are defined in the script called $globalVariables$, and it is important that the setting in simulink is to load global variables from this script. The scripts that model the kinetics, kinematics, and drag/lift coefficients are found in separate folders. Thees has to be added manually to the search path in matlab.

Bibliography

R. W. Beard and T. W. McLain. *Small unmanned aircraft.* Princeton University Press, 2012.

C.A.Marchaj. *Aero-hydrodynamics of sailing.* Tiller, 2000.

R. Carter. *A Companion to the Archaeology of the Ancient Near East*, chapter Watercraft. John Wiley & Sons, 2012.

B. Clement. Control algorithms for a sailboat robot with a sea experiment. Conference on Control Applications in Marine Systems, Osaka, Japan, 2013.

O. M. Faltinsen. *Sea loads on ships and offshore structures.* Cambridge university pass, 1990.

T. I. Fossen. *Handbook of Marine Craft Hydrodynamics and Motion Control.* WILEY, 2011.

E. Freund. *Decupling and pole assignment in nonlinear systems.* Wiley, 1973.

P. Herrero, L. Jaulin, J. Vehi, and M. A. Sainz. *Inner and outer approximation of the polar diagram of a sailboat.* 2005.

Y. Himeno. Prediction of ship roll damping-state of the art, research project report no. 239. *University of Michigan*, 1981.

S. F. Hoerner. *Fluid Dynamic Drag.* Hartford House, 1965.

A. Isidori. *Nonlinear control systems.* Springer-Verlag, 1989.

L. Jaulin and F. L. Bars. *Robotic Sailing 2012.* Springer Berlin Heidelberg, 2013a.

L. Jaulin and F. L. Bars. Sailboat as a windmill. *IRSC 2013 - 6th International Robotic Sailing Conference*, 2013b.

E. V. Lewis. *Principles of Naval Architecture, 2nd ed.* Society of Naval Architects, 1988.

T. Michael, P. Gilbert, C. Walsh, A. Bows, A. Filippone, P. Stansby, and R. Wood. Propulsive power contribution of a kite and a flettner rotor on selected shipping routes. *Applied Energy*, 2014.

NORSOK standard. *Actions and action effects (N-003).* Standards Norway, 2nd edition, 2007.

R. E. Sheldahl and P. C. Klimas. *Aerodynamic Characteristics of Seven Symmetrical Airfoil Sections Through 180-Degree Angle of Attack For Use In Aerodynamic Analysis of Vertical Axis Wind Turbines.* Sandia National Laboratories, 1981.

R. Stelzer, T. Proll, and R. John. Fuzzy logic control system for autonomous sailboats. Fuzzy Systems Conference, IEEE, 2007.

A. J. Sørensen. *Marine Control Systems, Propulsion and Motion Control of Ships and Ocean Structures, lecture notes in subject TMR4240.* NTNU, 2013.

A. Tiano, A. Zirilli, C. Yang, and C. Xiao. A neural autopilot for sailing yachts. page 27 – 29. in Proceedings of the 9th Mediterranean Conference on Control and Automation, 2001.

H. Wagner. *Planing of Watercraft.* NACA T.M. No. 1139. Washington, D.C., 1948.

K. L. Wille, V. Hassani, and F. Sprenger. Modeling and course control of sailboats. 10th IFAC Conference on Control Applications in Marine Systems, September 13-16, 2016a.

K. L. Wille, V. Hassani, and F. Sprenger. Roll stabilization control of sailboats. 10th IFAC Conference on Control Applications in Marine Systems, September 13-16, 2016b.

L. Xiao and J. Jouffroy. On motion planning for point-to-point maneuvers for a class of sailing vehicles. *Journal of Control Science and Engineering*, 2011.

L. Xiao and J. Jouffroy. Modeling and nonlinear heading control of sailing yachts. *Journal of Oceanic engineering*, 2014.

E. Yeh and J. bin. Fuzzy control for self-steering of a sailboat. volume 2, pages 1339 – 1344. Intelligent Control and Instrumentation, Singapore International Conference, IEEE, 1992.