Masterthesis

Simulation & Control Optimization of an autonomous sail boat

Fabian Jenne

Supervisor:
Dr. Cédric Pradalier
Jérôme Maye

Autonomous Systems Lab, Prof. Roland Siegwart
Swiss Federal Institute of Technology Zurich

SS 2010

Acknowledgment

First of all, I would like to thank my supervisors Cédric Pradalier and Jérôme Maye for their great support and the good introduction into C programming and for the advices as sailing experts. Another thanks goes to the Team SSA for the help and the good ideas concerning autonomous sailing. Further more, I would like to thank my fiancé Anastasia for carefully reading and correcting my thesis.

Abstract

Motion control systems have a significant impact on the performance of sail boats which act in severe sea conditions during long periods of time. This thesis focuses on real-time motion planing in a dynamic environment, particularly the Atlantic Ocean. It describes the design and implementation of an adaptive motion control for a complex sail boat system.

Based on a kinematic model, a Software-in-the-Loop (SiL) simulation will be established to analyze the system's behavior. Further, an optimized and compact rudder controller, which leads the vessel to its desired target will be realized. The main difficulty during that process is to tune the control values to ensure a robust and stable outcome.

The second part presents a global method for dynamic obstacle avoidance based on the Non-Linear Velocity Obstacle Concept. The rich information on the velocities admissible for the boat system is used to build a complete autonomous navigation module. It is composed of a local obstacle avoidance system, which is coupled with an incremental global motion planner.

Additionally, an error recovery software will be designed. Its objective is to detect sensor faults as well as instabilities of the system's performance. Fault detection is accomplished by evaluating significant changes in the behavior of the vessel with respect to the fault-free behavior estimation. Results obtained in simulations for dynamic environments will be presented and discussed.

Contents

List of Figures

1 Introduction

1.1 Motivation

Motion control for autonomous sail boats is an interesting research domain in robotics. For motorized vessels in an isotropic and stationary environment, where a straight line is the shortest path to the goal, both in terms of time and distance, the identification for an optimum heading to reach the target is trivial. This is significantly different for a sail boat, where a straight line path may not even be navigable if the goal is located upwind. In this case the sailor has to beat (sail a zic-zac course).

Apart from the autonomous platform, a sail expert can explain basic sailing skills with rules about how to steer the sail and the rudder according to direction of target and wind. The actual research is aiming to transform the sailor's knowledge into a control system and to provide a stable performance of the vessel. Robust behavior in reaching the target is a necessary condition for successful applications of autonomous sail boats.

The motivation for this thesis is the creative manner of controlling a complex system as well as implementing the theoretical software on a hardware system.

1.2 Background

In the previous year an autonomous sailboat has been designed with the objective to compete in the *Microtransat* Challenge [14]. Thus the vessel has to withstand the harsh conditions given on the Atlantic Ocean environment. The work consists of the mechanical and electrical realization of the boat as well as the software implementation that is able to compute an optimal path to reach a given target. Additionally, a controller was established to maneuver the vessel to its destination, taking into account the current environmental conditions, such as the wind speed or its direction.

1.3 Problem description

Initially it is assumed that the system, consisting of the mechanically designed sail boat and its sensors and actuators, depict in Figure 1, already exists. Further on, the above mentioned motion control software was established [6] leading the vessel from a given start point to a final destination point. During a real-time outdoor test, the controller could not prevent stability. Therefor the unstable sections have to be identified and further improved.

Figure 1: The autonomous sail boat AVALON

1.4 Contributions

The main contribution is to develop a mathematical model of the vessel within a simulation environment and to optimize the motion control system. Based on this, an error recovery and collision avoidance software can be established to guarantee safety during a long-term sail course. Fulfilling these tasks may involve the following steps:

- establish requirements for the system
- stating a kinematic model
- test, develop and extend the controller and path planing
- develop an error recovery software and a collision avoidance algorithm
- implement and test the controllers in the simulated environment

1.5 Overview of the thesis

The second chapter introduces the kinematic notations of the sail boat system which are fundamental for the rest of the work. In chapter 3 the design of the simulation platform will be described as well as an evaluation of the kinematic model. The next chapter handles the theory of motion control and shows an optimized rudder controller. Further on, an error recovery software will be developed. Chapter 5 presents a method for collision avoidance in a dynamic environment such as the Atlantic Ocean. Chapters 6 and 7 summarize the thesis and present the conclusions with an outlook for possible future works.

2 Sailboat Kinematics

2.1 Overview

Mathematical models of sailboat dynamics and disturbances are useful for computational simulations, for estimations of maneuvering qualities and for the prediction of the control design.
In the following section one specifies the kinematic model of a sailboat. With a linear model for the vessel's motion in deep and unrestricted waters the influence of wind, waves, and currents on the vessel's motion is discussed. The parameter values of the autonomous surface vehicle (ASV) are estimated from theoretical calculations and computer designed models. The purpose is to obtain a model which is suitable for the simulation of the vessel's steering.

2.2 Kinematic model of the sailboat

The ASV is considered as a rigid body with six degrees of freedom corresponding to translations along the three axes and its rotations about these axes. Thus, to receive a convenient expression for the equation of motion two types of reference frames have to be defined: the *inertial frame* and the *body-fixed frame*.

- **North-east-down frame (inertial frame):** This frame is fixed to the Earth. The positive x_n-axis points towards the North, the positive y_n-axis towards the East, and the positive z_n-axis towards the center of the Earth. The origin, o_n, is located on the mean water surface at an appropriate location. This frame is considered inertial. The assumption is reasonable since the vessel's velocity is small enough for the rotational force of the Earth being negligible compared to the hydrodynamic forces acting on the vehicle [3].

- **Body-fixed frame:** This frame is fixed to the hull. Figure 2 illustrates the definition of this coordinate system. As the symmetry of the hull can be exploited, the hydrodynamic forces and moments acting on the ship can easily be described in such a system.

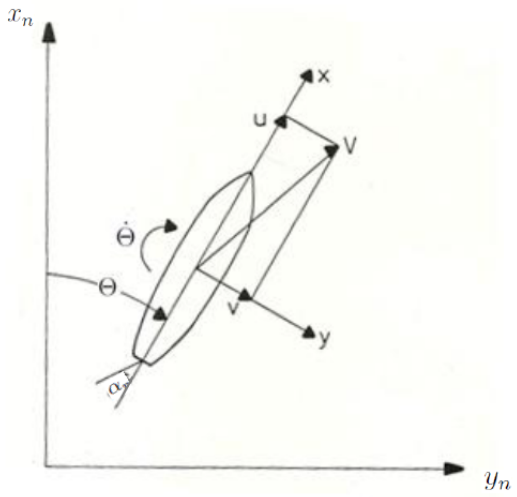

Figure 2: Notation and sign conventions for vessel motion description

To describe the position and the heading Θ of the ASV in the x, y-plane of the inertial frame, the pose $\vec{p}$ is introduced

$$\vec{p} = \begin{pmatrix} x \\ y \\ \Theta \end{pmatrix} \tag{1}$$

To simplify the equations describing the motion in the horizontal plane, the couplings between this motion and the heave, roll and pitch motions are neglected. This approximation is accepted since the coupling effects are insignificantly small.

The variables used to describe the horizontal motion are explained in Figure 3. The projections of the total speed V on the x- and y- axis are called the surge velocity u and the sway velocity v. The heading angle and the turning rate are denoted Θ and $\dot{\Theta}$.

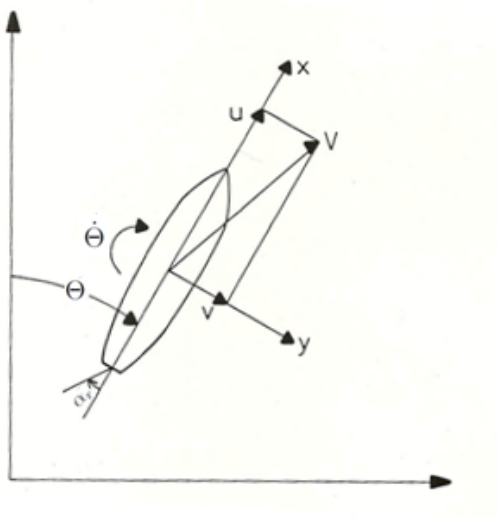

Figure 3: Variables used to describe the motion in the horizontal plane

With Newton's law expressing conservation of linear and angular moment the equation of motion can be obtained as following

$$\begin{aligned} m \cdot (\dot{u} - v \cdot \dot{\Theta} - x_G \cdot \ddot{\Theta}) &= X \\ m \cdot (\dot{v} + u \cdot \dot{\Theta} - x_G \cdot \ddot{\Theta}) &= Y \\ I_z \cdot \dot{\Theta} + m \cdot x_G \cdot (\dot{v} + u \cdot \dot{\Theta}) &= N \end{aligned} \tag{2}$$

where X and Y are the components of the hydrodynamic forces and disturbances along the x- and y-axis, N the z-component of the moments, m the mass of the vessel and I_z its moment of inertia. By setting the center-line of the vessel to the coordinate origin, the center of gravity x_G can be neglected.

2.3 Hydrodynamic Forces and Moments

To quantify the hydrodynamic forces used in the mathematical model (2), each single part of the vessel will be analyzed, beginning with the main construct, the sail.

2.3.1 Sail Forces

In order to establish the forces generated by the sail, the calculation of the apparent wind is required. It is the actual flow of air acting on the sail which differs in speed and direction from the true wind. It can be computed in a vector operation from the true wind angle γ_T and the vessel's velocity v as illustrated in the Figure below.

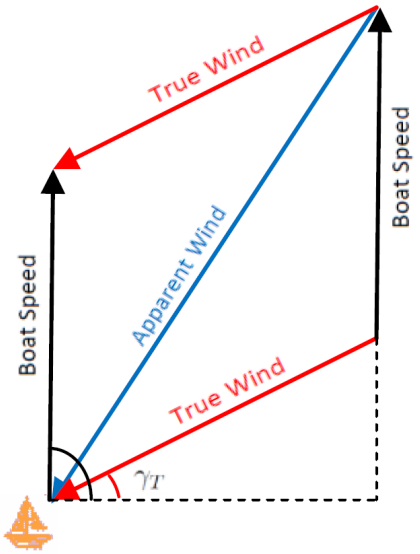

Figure 4: Definition of the apparent wind

Using the standard approximation for air foils in a moving fluid, the components of the sail force F_{sail} can be modeled as a combination of the drag force $F_{s,drag}$ and the lift force $F_{s,lift}$. The drag force is defined as the force in line with the onset flow and the lift force as being perpendicular to it

$$F_{s,lift} = 1/2 \cdot \rho_{air} \cdot v^2_{windapp} \cdot A_{sail} \cdot c_l(\alpha_{sail}) \quad (3)$$

$$F_{s,drag} = 1/2 \cdot \rho_{air} \cdot v^2_{windapp} \cdot A_{sail} \cdot c_d(\alpha_{sail}) \quad (4)$$

where ρ_{air} is the density of air, A_{sail} is the apparent area of the sail and c_l/c_d the lift/drag coefficient which depends on the sail's shape and the angle of attack α_{sail}.

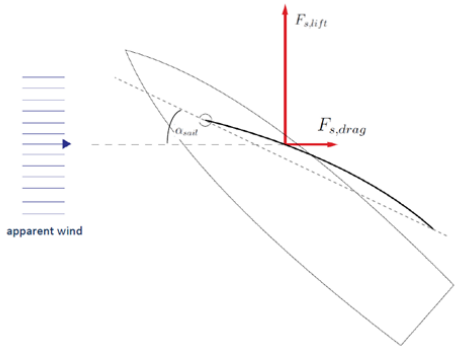

Figure 5: Drag and lift force acting on the sail

By rotation the force coordinate system through the apparent wind angle β the driving force $F_{s,R}$ and the heeling force $F_{s,H}$ can be obtained

$$F_{s,R} = F_{s,lift} \cdot \sin(\beta) - F_{s,drag} \cdot \cos(\beta) \quad (5)$$

$$F_{s,H} = F_{s,lift} \cdot \cos(\beta) + F_{s,drag} \cdot \sin(\beta) \quad (6)$$

The Figure below illustrates the transformation of the drag and lift force into the boat's coordinate system.

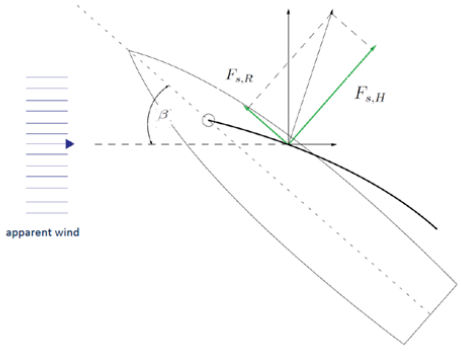

Figure 6: Driving and heeling force

The torque N_s generated by the driving and heeling force can be computed by estimating the position of the center of effort CE which depends on the angle of the sail with respect to the vessel.

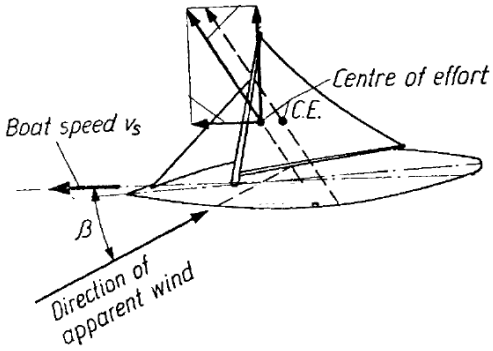

Figure 7: Definition of the center of effort CE

2.3.2 Rudder Forces

The system consists of two rudders. Due to the symmetric adjustment and an identical movement, the forces of the rudders are equivalent to each other. Analogical to the sail forces, the rudders act like wing foils in a moving fluid. The lifting action of the foil arises from the difference in the average pressure of the water over the upper and lower surface of the lifting rudder foil.

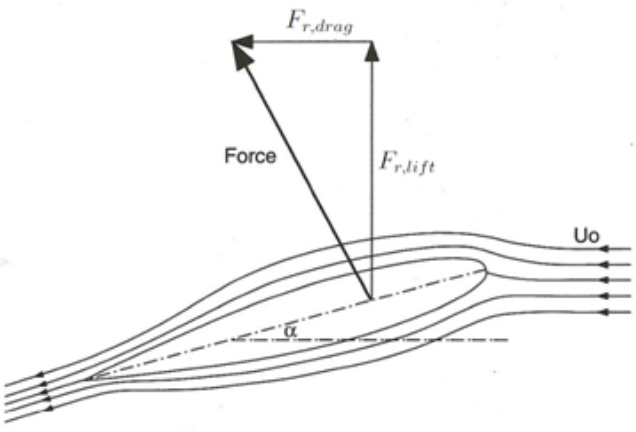

Figure 8: Pressures around a lifting rudder

Following normal practice, the total force is resolved into a lift component perpendicular to the water stream and a drag component parallel to it

$$F_{r,lift} = 1/2 \cdot \rho_{water} \cdot v^2_{r,waterapp} \cdot A_{rudder} \cdot c_l(\alpha_{rudder}) \quad (7)$$

$$F_{r,drag} = 1/2 \cdot \rho_{water} \cdot v^2_{r,waterapp} \cdot A_{rudder} \cdot c_d(\alpha_{rudder}) \quad (8)$$

where ρ_{water} is the density of water, A_{rudder} is the apparent area of the rudder and c_l/c_d the lift/drag coefficient depending on the rudder's shape and the angle of attack α_{rudder}. Figure 9 shows the characteristics of the lift and drag coefficient against the incidence angle α_{rudder}. Initially $c_l(\alpha_{rudder})$ rises nearly linearly with a characteristic lift slope and then more steeply until the *stall point* is reached. At this point the curve flattens and ultimately falls rapidly, after which the foil is said to be stalled. The drag coefficient $c_d(\alpha_{rudder})$ rises approximately parabolically with incidence until near stall where a significant increase in the rate of rise occurs.

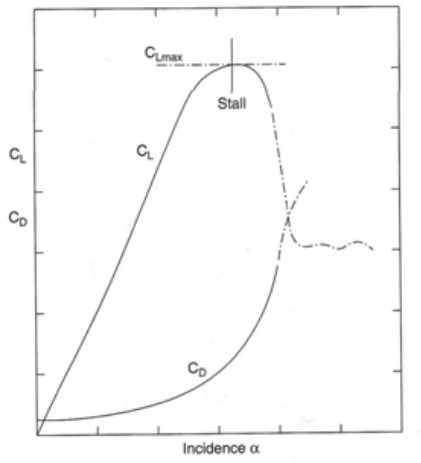

Figure 9: Lift and Drag characteristics of the rudder

To calculate the apparent water velocity $v_{r,waterapp}$ acting on the rudders, the rotation of the vessel around its center has to be considered

$$\vec{v}_{r,waterapp} = \vec{v}_{waterapp} + \vec{\Omega} \times \vec{r} \quad (9)$$

where $\vec{r}$ expresses the vector distance of the rudder center to the vessel's center and $\vec{\Omega}$ the rotation speed of the ASV.
Equivalent to the sail forces a transformation through the apparent water angle leads to the rudder's driving and heeling forces $F_{r,R}$ and $F_{r,H}$ generated by the rudder's lift and drag forces.

2.3.3 Damping Forces

Based on the geometrical shape of the vessel's components, several damping forces have to be considered. Mainly these forces are based on the resistance of the hull and keel along the axes of translational freedom. They depend on the velocity and the rate of turn of the ASV and are modeled using the equation

$$\vec{F}_{damp} = d \cdot \vec{v}^2 \tag{10}$$

where $\vec{v}$ represents the velocity or rate of turn and d the damping parameter identified experimentally [6].

2.3.4 Disturbances

The mathematical model of (2) does not consider disturbances. However, a vessel is influenced by wind, waves and currents. They appear as disturbances in the equation of motion. It is assumed that the disturbances can be modeled by forces and moments. This is an approximation because the superposition principle does not necessarily hold for large maneuvers [2].
The following disturbances will be discussed:

- Current
- Wind
- Waves

Current

The current is assumed to be constant and homogeneous. It influences only the inertial velocity of the ASV, but not the hydrodynamic forces and moments. Relating to Figure 10 the velocity components of the current are obtained as following

$$\begin{aligned} u_c &= V_c \cdot \cos(\gamma_c - \Theta - \pi) \\ v_c &= V_c \cdot \sin(\gamma_c - \Theta - \pi) \end{aligned}$$

with V_c denoting the current velocity and γ_c the current direction.

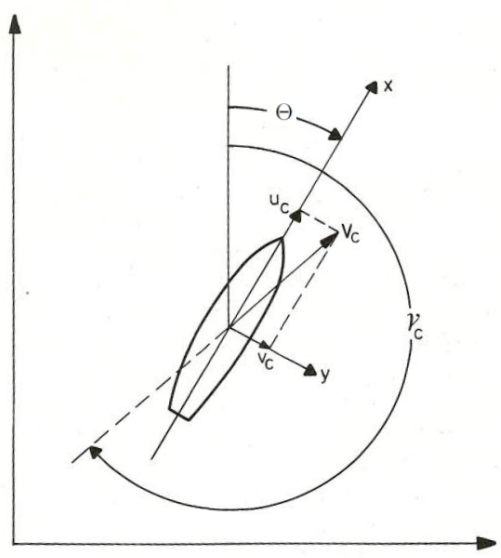

Figure 10: Definition of the velocity and the direction of the current

Thus, the influence of the current on the vessel's motion is given by forces and moments

$$\begin{aligned} X_{current} &= m \cdot v_c \cdot \dot{\Theta} \\ Y_{current} &= -m \cdot u_c \cdot \dot{\Theta} \\ N_{current} &= -m \cdot x_G \cdot u_c \cdot \dot{\Theta} = 0 \end{aligned}$$

which are added to the kinematic model.

Wind

Wind can be regarded as a combination of two elements, the mean wind and the turbulent wind. The mean wind is a constant, homogeneous air-current. The turbulent wind is considered as a stochastic fluctuation. As the area of the hull above the water surface is small, only influence of the mean wind is considered. Based on [2] the following expressions is used

$$\begin{aligned} X_{wind} &= 1/2 \cdot \rho_{air} \cdot V_R^2 \cdot c_X(\gamma_R) \cdot L^2 \\ Y_{wind} &= 1/2 \cdot \rho_{air} \cdot V_R^2 \cdot c_Y(\gamma_R) \cdot L^2 \\ N_{wind} &= 1/2 \cdot \rho_{air} \cdot V_R^2 \cdot c_N(\gamma_R) \cdot L^3 \end{aligned}$$

where ρ_{air} is the density of air ($\rho_{air} = 1.23 kg/m^3$) and L the vessel length. The wind direction γ_R as well as the wind speed V_R are explained in Figure 11.

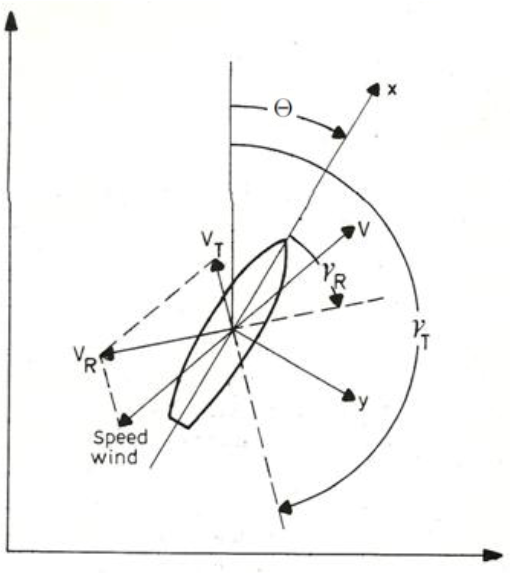

Figure 11: Definition of the wind speed and direction

The relative wind components are related to the true wind components through the following identities:

$$
\begin{aligned}
V_R &= \sqrt{u_R^2 + v_R^2} \\
\gamma_R &= \arctan(v_R/u_R) + \pi \quad \text{if} \quad u_R >= 0 \\
\gamma_R &= \arctan(v_R/u_R)(+2\pi \quad \text{if} \quad v_R > 0) + \pi \quad \text{if} \quad u_R < 0
\end{aligned}
$$

whit

$$
\begin{aligned}
u_R &= V_T \cdot \cos(\gamma_R - \Theta - \pi) - u - u_c \\
v_R &= V_T \cdot \sin(\gamma_R - \Theta - \pi) - v - v_c
\end{aligned}
$$

In recent works the coefficients c_X, c_Y and c_N have been determined for different types of vessels [1].

Waves

Ocean waves are random in both time and space. To find a reasonable characterization of the waves, a simplified model describing regular waves is used. It considers a simple two-dimensional wave train progressing over an infinite water surface with an infinite depth. This approximation is commonly referred to as a regular sea.
Assuming the wave amplitude is small compared with the wavelength and the water is incompressible, a wave profile may be obtained using the following expression

$$\xi(x_0, t) = \frac{h}{2} \cdot \cos(kx_0 - \omega t) \tag{11}$$

where $\xi(x_0, t)$ is the z_0 coordinate of the sea level at x_0 at time t, and h the wave height, depict in Figure 12. The wave number k is defined as $k = 2\pi/\lambda$ with the wavelength λ. The theory of gravity waves gives the relation

$$k = \frac{4\pi^2}{gT_w^2}$$

with g as the acceleration of gravity and T_w the wave period.

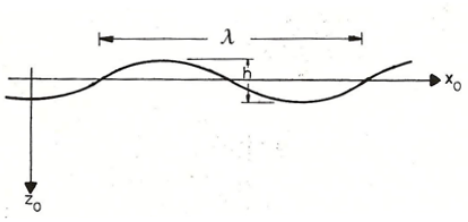

Figure 12: A sinusoidal wave with length λ and height h

The wave slope $s(x_0, t)$ can be obtained from (11)

$$s(x_0, t) = \frac{\partial \xi}{\partial x_0} = -(k \cdot h/2) \sin(kx_0 - \omega t) \tag{12}$$

To derive an expression for the forces and moment induced by the waves, two assumptions are made additionally:

- the forces and moment only result from pressure
- the wave field is not disturbed by the presence of the vessel

Thus, the following equations for forces and moment can be calculated to

$$
\begin{aligned}
X_{waves} &= 2 \cdot a \cdot B \cdot \frac{\sin(b) \cdot \sin(c)}{c} \cdot s(t) \\
Y_{waves} &= -2 \cdot a \cdot L \cdot \frac{\sin(b) \cdot \sin(c)}{b} \cdot s(t) \\
N_{waves} &= a \cdot k \cdot \left[B^2 \cdot \sin(b) \cdot \frac{c \cdot \cos(c) - \sin(c)}{c^2} \right. \\
&\qquad \left. + L^2 \cdot \sin(c) \cdot \frac{b \cdot \cos(b) - sin(b)}{b^2} \cdot \xi(t) \right.
\end{aligned}
$$

with

$$
\begin{aligned}
a &= \rho_{water} \cdot g \cdot (1 - e^{-k \cdot T})/k^2 \\
b &= k \cdot L/2 \cdot \cos(\chi) \\
c &= k \cdot B/2 \cdot \sin(\chi)
\end{aligned}
$$

and

$$
\begin{aligned}
s(t) &= s(0,0,t) = \frac{k \cdot h}{2} \sin(\omega_e t) \\
\xi(t) &= \xi(0,0,t) = \frac{h}{2} \cos(\omega_e t)
\end{aligned}
$$

For low and moderate sea states, the sea can be considered stationary for periods over 20 minutes. At medium state wavelengths are usually of the order of $50 - 150m$ [3].

2.4 Forward Kinematic Representation

The kinematic model given by eq.(2) changes with the introduction of the hydrodynamic forces explained in Chapter 2.3. Additionally in the presence of wind, waves, and currents the equation of motion can be written as

$$\begin{aligned} m \cdot (\dot{u} - v \cdot \dot{\Theta}) &= X_{tot} \\ m \cdot (\dot{v} + u \cdot \dot{\Theta}) &= Y_{tot} \\ I_z \cdot \dot{\Theta} &= N_{tot} \end{aligned} \tag{13}$$

with

$$\begin{aligned} X_{tot} &= F_{s,R} + F_{r,R} + F_{damp,x} + X_{current} + X_{wind} + X_{waves} \\ Y_{tot} &= F_{s,H} + F_{r,H} + F_{damp,y} + Y_{current} + Y_{wind} + Y_{waves} \\ N_{tot} &= N_s + N_r + N_{damp} + N_{current} + N_{wind} + N_{waves} \end{aligned} \tag{14}$$

This mathematical model allows one to design a control system to perform numerical simulations of different scenarios, and to obtain a preliminary assessment of the impact the design can have on the performance of the system. Therefor a discretized velocity model is obtained. Given the initial position and velocity as well as the sampling time Δt, the velocities at time $(k+1)$ can be computed to

$$\begin{aligned} \dot{\Theta}_{k+1} &= \frac{N_{tot}}{I_z} \cdot \Delta t + \dot{\Theta}_k \\ \dot{y}_{k+1} &= \frac{1}{1 + \dot{\Theta}_{k+1} \cdot \Delta t^2} \cdot (\frac{Y_{tot}}{m} \cdot \Delta t + \dot{y}_k - \frac{X_{tot}}{m} \cdot \Delta t^2) \\ \dot{x}_{k+1} &= (\frac{X_{tot}}{m} + \dot{y}_{k+1} \cdot \dot{\Theta}_{k+1}) \cdot \Delta t + \dot{x}_k \end{aligned} \tag{15}$$

and the position update, described in the inertial frame, is defined as

$$\vec{p}_{k+1} = \vec{p}_k + \bar{R}_{\Theta,z} \cdot \vec{v}_{k+1} \cdot \Delta t \tag{16}$$

where $\vec{v}_{k+1}$ represents the velocity vector and $\bar{R}_{\Theta,z}$ a rotational transformation matrix around the z-axis from the boat frame into the inertial frame with angle Θ.
It has to be emphasized that the equations of motion are only approximations. They form, however, a suitable model for the computational simulation of steering which will be described in the next chapters.

2.5 Sail states

The kinematic constraints limit the maneuverability of the vessel. Thus, the sailboat can not sail at all angles to the wind. Figure 13 shows the range that can not be sailed, represented in gray. In order to achieve high velocities, it has to be differentiated between several states of sailing. To do so, a controller is designed as a state machine [6], able to switch the states depending on the apparent wind angle. The possible sailing states are illustrated in Figure 13 and define the areas in where the vessel is allowed to sail with respect to the wind.

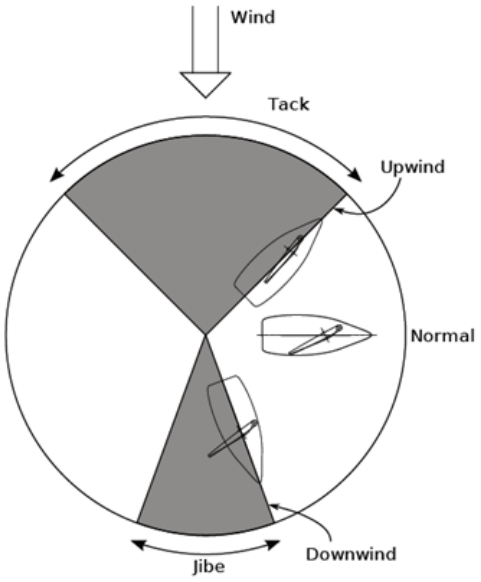

Figure 13: Possible courses of a sailing yacht with respect to the wind

3 Simulation Design

3.1 Overview

This chapter presents the Matlab/C++ based Software-in-the-Loop simulation (SiL), which integrates the control system in a collective simulation environment. After the verification of the motion behavior, the controller can be optimized and will then be implemented on the real system.

3.2 Architecture

To establish a reliable behavior of the ASV a simulation platform is designed. The objectives are to reproduce an artificial sensor data set similar to the ones occurring during a sail course and to observe the vessel's movements. With additional informations about the rudder and sail states and their resulting forces the controller can be optimized. The basic concept of the SiL architecture is illustrated in the Figure below.

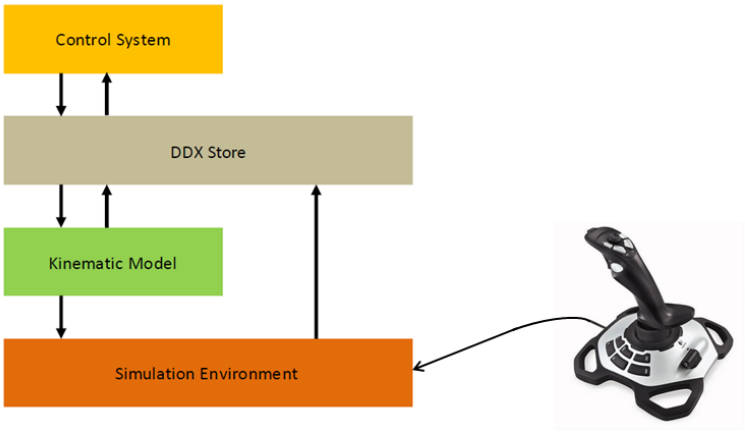

Figure 14: General architecture of the SiL

This concept consists of four main parts and their connection, representing the communication between the programs.

DDX

The base of the entire software structure is the middleware software DDX [7]. This software manages a shared memory called the *store* and thus provides an appliance of communication between the individual programs. The principles of DDX allows to structure the control system in closed independent parts which are only connected through the store by reading and writing global variables. For instance, sensor drivers read out actual sensor data from the hardware and write it to the store. Any other program that requires the sensor data can now read the data from the store. The distinct advantage of DDX is its modular software structure. It provides the opportunity to restart a crashed program without interrupting the other running processes.

Control System

To reach a desired target, with the given initial coordinates of the ASV, a control system is designed [6]. First a trajectory planer is used to generate an optimal path. The planing algorithm is split into a global and a local planning. In the global phase a rough mapping from the current vessel position to the final destination is computed. Based on the rough mapping a detailed path can then be produced with the local planer, incorporating only the next few nautical miles along the rough path.
The next stage is to follow the given trajectory. Therefor a sail control and rudder control are established. Both operate separately with two Single Input Single Output (SISO) systems, assuming they do not correlate with each other.

The following Figure represents the general software structure.

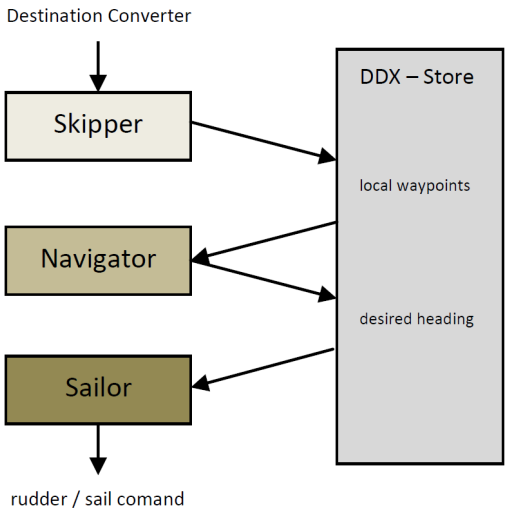

Figure 15: Principle of the control system structure

At the very beginning the destination coordinates will be converted and sent to the *Skipper* program. Inhere, local waypoints are generated which lead the vessel to its target. With the *Navigator* an optimal path is planed and the corresponding desired heading is set to the store. With the information concerning the heading, the *Sailor* controlls the sail and rudder position to steer the vessel along the path.

The *sail control* outputs an optimal angle of attack α_{sail} of the vessel's sail, based on the desired heading, and the current wind informations. The optimal angle is derived from a predefined value that was found in experiments. For safety reasons, a dynamic change of the sail angle based on the wind speed is necessary. In general, as the wind increases, the desired α_{sail} approaches zero to reduce the aerodynamic sail forces generated by the wind. With this condition the vessel remains steerable even in strong winds.

The *rudder control* keeps track of the vessel's heading. Even though a dynamic heading change is not directly dependent on the rudder angle, a PID controller is designed to regulate the rudder position. In chapter 4.3 an optimized rudder and sail controller will be developed with the objective to gain stability.

While steering a sail boat, leeway drift has to be compensated. Therefor the vessel has to sail closer to the wind than the desired heading $\Theta^{\star}$ predicts. With the sensor data the drift velocity v_{drift} can be accurately measured and a new desired heading is calculated to

$$\Theta^{\star}_{opt} = \Theta^{\star} + \arctan(\frac{v_{drift}}{v_{boat}}) \tag{17}$$

with

$$v_{drift} = \sqrt{v_y^2 + v_z^2} \cdot \ \text{sign}(\varphi_x) \tag{18}$$

where φ_x represents the angular position of the vessel around the x-axis, called roll angle.

The kinematic model design refers to the forward kinematic representation described in chapter 2.4.

Simulation Environment

Figure 16 depicts the visualization of the simulation environment. It includes the representation of the current state of the ASV with its heading, the rudder and the sail state as well as the driven trajectory. Further more, informations about the desired heading, the speed and direction of the wind, waves, and current are given. Additionally the behavior of the different sensors is illustrated.

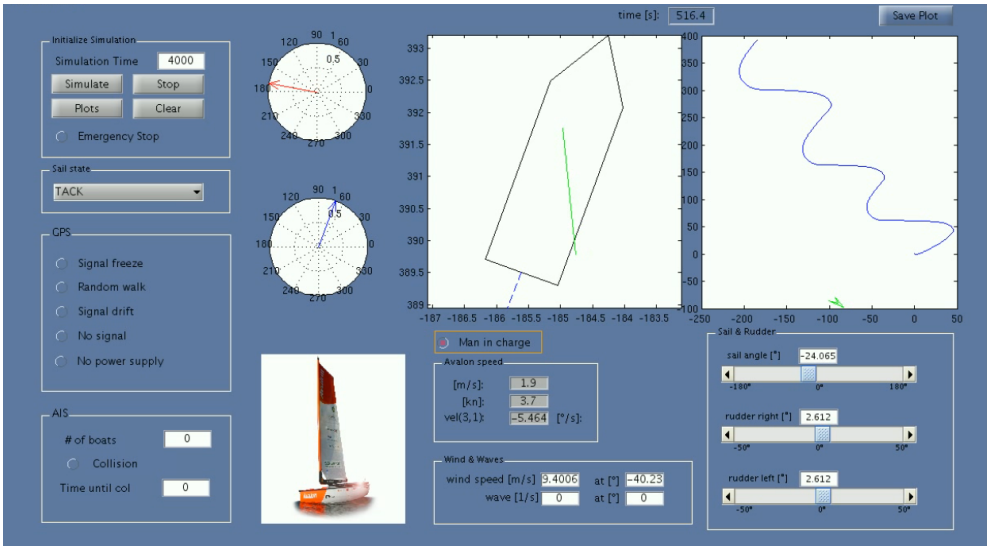

Figure 16: The simulation environment

3.3 System Identification

Logged data from an outdoor real-time course is used to evaluate the modeled system and to adjust the parameters. Data such as the observed wind, the resulting velocities and the passed trajectory of the real system is regarded. By simulating an equivalent path with equivalent wind data, the system parameters of the modeled vessel are adjusted. Figure 17 depicts a trajectory of the simulated model (dashed-line), compared with a trajectory of the logged files. Thus, the system is identified and a similar behavior of the simulation compared to the the real hardware is established.

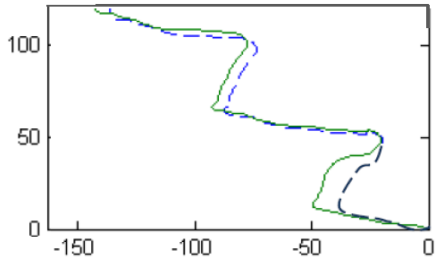

Figure 17: Trajectory of the simulation (dashed-line) and the real vessel

The following Figure presents the corresponding velocities in x, y and z directions of a simulated path. It can be seen that the maximum speed in the longitudinal direction of the boat is approximately 6 knots and that the drift velocity in the y direction is rather small. This behavior is also obtained on the real system.

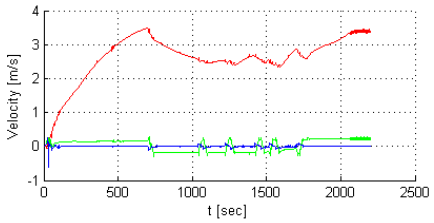

Figure 18: Corresponding velocities of the vessel in x, y and z direction

4 Motion Control

4.1 Overview

Based on the kinematic model of the ASV, described in chapter 2.4 and the designed simulator, the controller of chapter 3.2 has to be analyzed. With a specific *spiral test* an instability of the rudder control will be determined. Later on in this chapter, an optimized controller will be introduced as well as an error recovery software with the main focus on identifying an error that may occur during a long-term course.

4.2 Performance analysis

The controller introduced in chapter 3.2 provides an unstable behavior. As depict in Figure 19, when turning the ship (area in the red circle), the rudder was not capable of placing and holding the vessel at an angle of attack to the flow of water past the hull. An overshoot in the amplitude of the rudder angle lead to an unexpected turning rate, unpromising to capture a stable position.

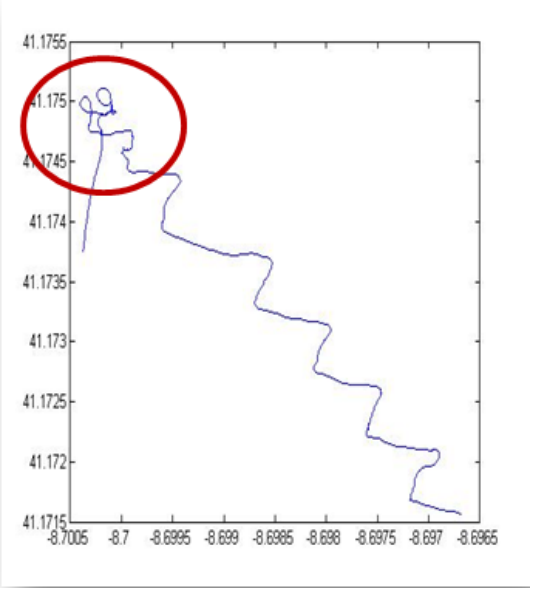

Figure 19: Instability of the rudder control

A *spiral test* is used to assess the directional stability or instability of a ship [3]. Therefor the vessel is simulated as a motor ship, generating a constant force in the longitudinal direction where the sail is neglected. On a steady approach, the rudder angle is set over to $+15°$, and the ship is allowed to settle to a steady rate of change of heading. The angle is then reduced incrementally until it reaches $-15°$ and the whole procedure then repeated back to $+15°$. The results obtained from this test are plotted in Figure 20 as rate of change of heading with respect to the rudder angle. In the case of a stable manoeuvre, the curve should be a single function, but as for the unstable controller from chapter 3.2 the curve has the appearance of a hysteresis loop (dashed-line curve). The larger the loop the more unstable the vessel.

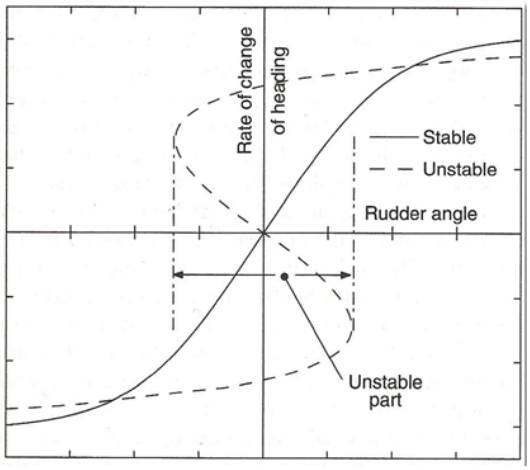

Figure 20: Spiral test

As the controller form the chapter 3.2 is of wrong order, it can not be modified to gain stability. Hence, a new controller must be established to guarantee a stable and robust steering behavior during a sail course.

4.3 Control optimization

To prevent the system from instability a new controller is designed. The vessel is declared to be directionally stable if, after deflected from its straight-line path, by say wind or waves, it returns to a new straight-line path, even though this might not necessarily be in the same direction as the original one. For a long time cruse the objective is to gain a high measure of directional stability. This will result in good coursekeeping, but if to high, will lead to low ability to manoeuvre.
The old controller adjusted the rudder based on the direct error in the heading. The main idea of the new controller is to stabilize the rotation speed of the vessel.

Based on the direct error in the vessel's heading, a PID controller is used to compute the desired rotation speed $\dot{\Theta}^{\star}$ the vessel needs to produce.

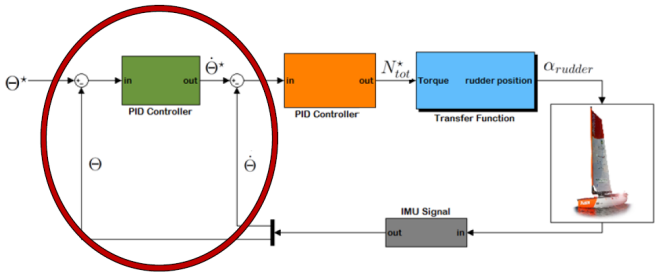

Figure 21: PID controller (red circle) to compute the desired $\dot{\Theta}^{\star}$

With the controller's output and the sensor's measurement of the rotation speed $\dot{\Theta}$, an additional PID controller is added to compute the desired torque $N^{\star}_{tot}$ for the vessel, necessary to reach the desired rotation speed. In order to avoid dealing with an overshoot of the rudders, the first controller is adjusted with a higher derivative term.

The derivative term reduces the rate of change of the output. Hence, it is used to decrease the magnitude of the overshoot generated by the integral component and to improve the process stability.

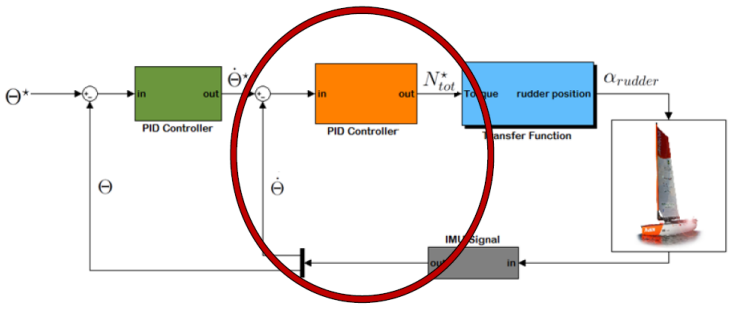

Figure 22: 2^{nd} PID controller to compute the desired torque $N^{\star}_{tot}$

With a given desired torque, the required rudder position has to be clarified. Therefor, a transfer-function must be established declaring the relations between the rudder position and the according torque of the vessel. With eq.(14), defining the total torque N_{tot} of the vessel, the rudder torque N_r can be computed

$$N_r = f(N^{\star}_{tot}, n(s)) \tag{19}$$

where $n(\overline{s})$ is a function describing the torques of the other geometrical parts. The function $n(\overline{s})$ depends on the active state of the wind $\overline{s}_{wind}$, the current $\overline{s}_{current}$, the waves $\overline{s}_{waves}$ and the vessel's state $\overline{s}_{boat}$.

$$n(s) = f(\overline{s}_{wind}, \overline{s}_{current}, \overline{s}_{waves}, \overline{s}_{boat}) \tag{20}$$

With the combination of eq.(19) and eq.(7) a relation between the rudder angle α_{rudder} and the rudder torque N_r is found and therefor a first order transfer-function can be computed

$$\alpha_{rudder} = f(N^{\star}_{tot}, n(s)) \tag{21}$$

In order to solve the equation it is reformulated into a problem that can be solved numerically through an iterative method.

$$0 = f(N^{\star}_{tot}, n(s)) - \alpha_{rudder} \tag{22}$$

With a root-finding algorithm [15] and an expedient initial value the necessary α_{rudder} for a given $N^{\star}_{tot}$ can be computed. Figure 23 shows the plot of eq.(22) for a given $N^{\star}_{tot}$.

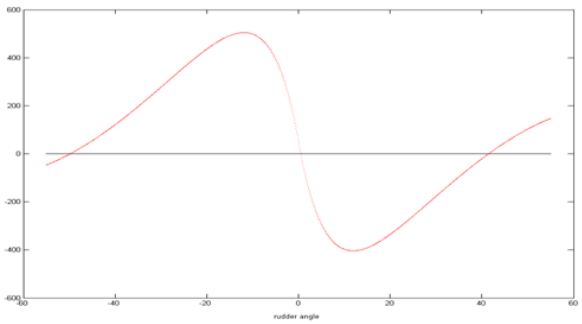

Figure 23: Plot of the transfer function for a given Torque

The initial values $\alpha_{1,2}$ used to solve this problem are chosen to be in the range close to the apparent water angle at the rudders

$$\alpha_{1,2} = \alpha_{r,waterapp} \pm 7^{\circ} \tag{23}$$

with the apparent water angle

$$\alpha_{r,waterapp} = atan2(v_{rwaterapp,y}, v_{rwaterapp,x}) \tag{24}$$

The iterative problem can only be solved if eq.(22) is monotonic. The derivative shows, that within a given rudder position this condition is fulfilled.

At low speed (< 0.5 knots) the range of the rudders limit the maneuverability and it may happen that a heading change can not be fulfilled. Hence eq.(22) does not cross the zero line and the algorithm can not find a root. To ensure the vessel still follows the correct path, an optimal rudder angle is found at the minimum of eq.(22), depict in the Figure below.

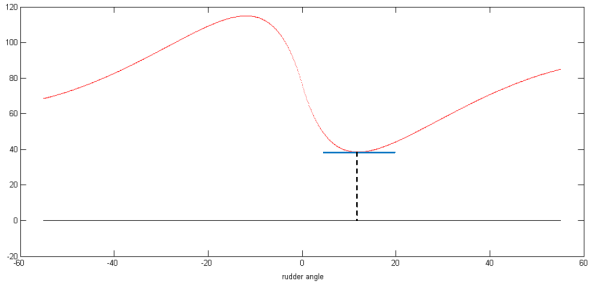

Figure 24: Plot of the transfer function at low speed

Figure 25 illustrates the summarized process of the new controller. It consists of two PID controllers and the aforesaid transfer-function.

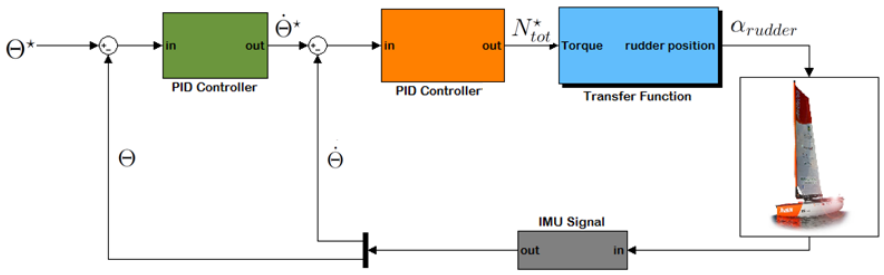

Figure 25: Illustration of the rudder control process

The tuning of the controller's parameter is done manually. To do so, the first controller is replaced by a rotation speed-profile and the second controller is tuned until a reasonable output trajectory is found.

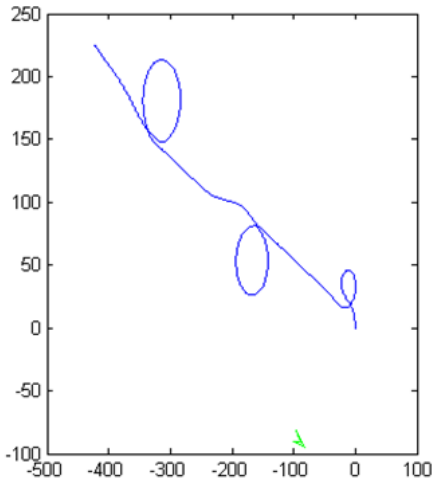

Figure 26: Trajectory for a given rotation speed-profile

In the next step a desired heading-profile is given and the first controller can be tuned. Not to have any overshoot of the rudder position and the rotation speed, a robust and slow controller is preferred. This yields to a slower turning rate during a tack or a jibe manoeuvre but prevents stability even at high speed. The following figure compares the desired and the resulting rotation speed during a zic-zac course. It can be seen that no overshooting occurs, and that the controllers smoothly follow the desired rotation speed.

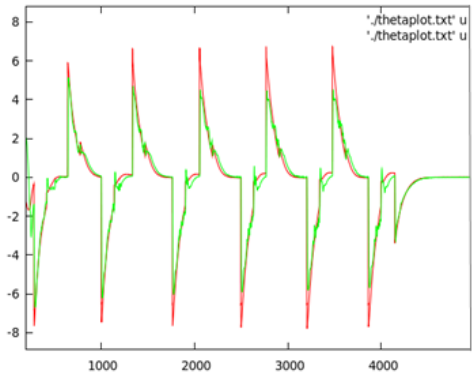

Figure 27: Desired and resulting rotation speed during a zic-zac course

4.4 Error Recovery Software

Identification of robust control must deliver not only a nominal model, but also a reliable estimate of the uncertainty and the probable errors associated with the model. During a sea trial several errors may occur. These errors can be classified as different failure types. The purpose of this section is first to simulate and determine the error and later on to handle it by obtaining prevention.
Three different failure types will be analyzed:

- Sensor-signal errors
- Failures, based on not running programs
- System-stability errors

In standard identification problems, the signal error originates from two different sources: a *variance* term adapted from noise affecting the data and a *bias* term based on system dynamics which is not captured by the estimated model. Clearly, the nature of these two error terms is different: the former is generally uncorrelated with the input signal while the latter strongly depends on the estimated model and on the input used in the identification.
The sensor signals are modeled discrete-uniformly distributed. This discrete probability distribution can be characterized by assuming that all values of a finite set of possible values are equally distributed.

The first failure type addresses an unexpected output of the sensors. Sensor validation refers to determination of whether a sensor is providing a correct signal. With focus on the GPS and the IMU sensor, different type of errors are simulated:

- *signal freeze*: the signal remains for a time sequence at the last measured state
- *random walk*: the signal of the sensor outputs wild points in an unexpected direction
- *zero-output*: the signal communication is not working and the sensor output is equal to 0

To determine the error based on incorrect sensor signals, a set of signal outputs is considered for a specific time range $[t_a, t_b]$. Further on, independent residuals are constructed for each different sensor failure. This residuals

are designed so that they correspond to an individual failure. With the knowledge of the dynamic system a rough prediction can be established, determining a range in which the sensor signal must lie for a given time step. Figure 28 illustrates an example that determines an error in the position of the boat. By knowing the maximum speed the vessel can gain, the maximum distance between two measured position points is defined. Therefor the range for the next measurement is specified (red circle). If the residual between two measurements exceeds this range, an error of the sensor signal is detected.

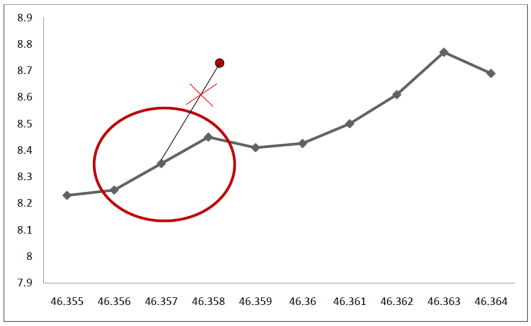

Figure 28: Measurement range definition to determine signal errors

Additionally, if the measurements stay within a small range (green circle) for a longer time a signal freeze is determined.

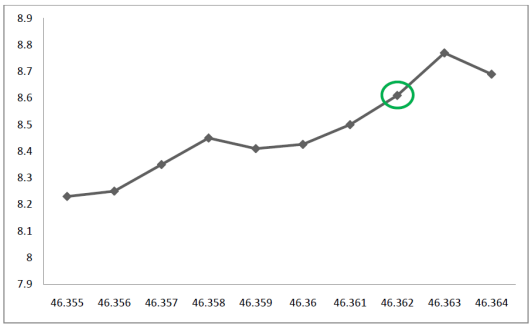

Figure 29: Measurement range definition to determine a signal freeze

The second failure type describes situations where on-board computer programs are not running. To determine this type of failures the assumption is made that the DDX store, described in the chapter 3.2, is permanently stable. By introducing a counter to the store, initialized with a specific value counting backwards, each time a program performs one step it sets the value in the store back to its initial condition. If a program is not running anymore, the variable in the store is counted backwards. At the stage it reaches

the value equal to zero the program is identified as not running anymore. In this case it has to be rebooted.

The last failure type analyzes the system's stability. An instability in the movement is recognizable in the vessel's rotation speed $\dot{\Theta}$. The system identification is performed by first predicting the rotation speed for the next time step $\dot{\Theta}_{k+1}$ and then comparing it with its observation using the IMU sensor. Generally the pose of the ASV is represented by the vector

$$\vec{p} = \begin{pmatrix} x \\ y \\ \Theta \end{pmatrix} \tag{25}$$

Starting from a known position, the position of the vessel can be estimated by integrating the movement (summing the incremental travel distances). For the discrete system with a fixed sampling interval Δt the incremental travel distances $(\Delta x, \Delta y, \Delta\Theta)$ can be computed using eq.(16). Thus we get the updated position $\vec{p}_{k+1}$. Similarly using eq.(15) the rotation speed is predicted to be

$$p(\dot{\Theta}_{k+1} \mid \dot{\Theta}_k, u_k(f(N))) = \frac{N}{I_z}\Delta t + \dot{\Theta}_k \tag{26}$$

Using the IMU sensor an observation of the rotation speed at time k is measured

$$p(\dot{\Theta}_k \mid z_k) \tag{27}$$

where z_k represents the sensor measurement.
To parse the systems behavior the error e_{SI} of the prediction step and its observation over a specific time range is analyzed

$$e_{SI} = \frac{1}{n}\sum_{n=0}^{N} \left\| p(\dot{\Theta}_{k+1} \mid \dot{\Theta}_k, u_k(f(N))) - p(\dot{\Theta}_k \mid z_k) \right\| \tag{28}$$

If the mean value exceeds a specific threshold the system is determined to be unstable in its rotation speed.

Consider the case where a fisher net blocks one of the two rudders. In this case the blocked rudder is not able anymore to provide the necessary torque for the vessel to reach its desired heading. An error in its rotation speed will be detected. Further, the current state of the rudder is compared to the desired rudder position and the blocked rudder is located. With this information one supposes that the blocked rudder is at a constant position. Therefor the moment of this blocked rudder can be established and will be added to the equation (14) to calculate the resulting torque

$$N_{tot} = N_s + N_{r,blocked} + N_r + N_{damp} + N_{current} + N_{wind} + N_{waves} \tag{29}$$

and the transfer-function from chapter 4.3 tries to find an optimal rudder position for the working rudder.

5 Dynamic Obstacle Avoidance

5.1 Overview

This section introduces the Automatic Identification System (AIS) and its use to the ASV in a ship populated environment such as the Atlantic Ocean. Motion planing in a dynamic uncertain environment is a difficult problem, since it requires planning in the state-space, that is simultaneously solving the path planning and the velocity planning problems:

- Path planning is a kinematic problem which involves the computation of a collision-free path from a start to a goal position
- Velocity planning requires the consideration of robot dynamics and actuator constraints

5.2 AIS Sensor

The objectives for implementation of the AIS are to enhance safety and efficiency of navigation and maritime environmental protection. It has been done through better identification of vessels, assisted target tracking, and improved situational awareness. The AIS information consists of different types of data, classified as static, dynamic, voyage related information and short safety messages. The dynamic data will automatically be updated through the AIS-connected ship sensors, and voyage related data will be entered manually during each voyage. The following information is included in the AIS messages:

- Static information:
  - Maritime Mobile Service Identity number (MMSI)
  - Type of vessel (passenger, tank, etc.)
  - Length and beam
  - Location of position fixing antenna such as GPS/DGPS
- Dynamic information
  - Ship's position with accuracy indication and integrity status
  - Time in UTC (coordinated universal time)
  - Course over ground (COG)

- Speed over ground (SOG)
- Heading
- Navigational status (e.g., not under command, constrained by draught, etc.)
- Rate of turn
- Angle of hell (optional)
- Pitch and roll (optional)

- Voyage related information
  - Ship's draught
  - Type of cargo
  - Destination and estimated time of arrival
  - Route plan-waypoints
  - Number of persons on board (on request)
- Short safety messaging

5.3 The Velocity Obstacle

With the information of the AIS sensor a first order method can be determined to compute the trajectory of the vessel moving in a time-varying environment. With the use of velocity information potential collisions can be predicted. The method consists of selecting avoidance maneuvers to avoid static and moving obstacles in the velocity space, based on the current positions and velocities of both the ASV and the obstacles. It is a first order method, since it does not integrate velocities to yield positions as functions of time.

For simplification, the analysis is restricted to circular objects, thus considering a planar problem with no rotations. This is not a severe limitation, as general polygons can be represented by a number of circles [21]. Further, one assumes that the obstacles move along arbitrary trajectories and that their instantaneous state, consisting of its position and velocity, is either known or measurable. In this case the AIS measurements are used for this informations. The speed of the ASV can be obtained using the IMU sensor. The direction of the speed vector is assumed to be the equivalent to the heading of the vessel to the current global waypoint. This direction is not similar to

the current heading, as the current heading might not point into the direction the ASV wants to sail. The following Figure illustrates the situation during a zic-zac maneuver where the current heading is not equivalent to the direction to the waypoint.

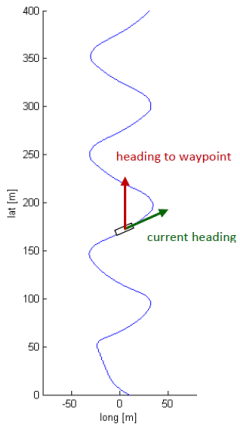

Figure 30: Vectors of the current heading and of the direction to the target

Consider the two circular objects, A and B, shown in Figure 31 at time t_0, with velocity v_A and v_B. Circle A represents the ASV and circle B the obstacle. To compute the *velocity obstacle* (VO), B has to be mapped into the configuration space of A. This is done by reducing A to the point $\hat{A}$ and enlarging B by the radius of A to $\hat{B}$. The state of each moving object is represented by its position and a velocity vector attached to its center.

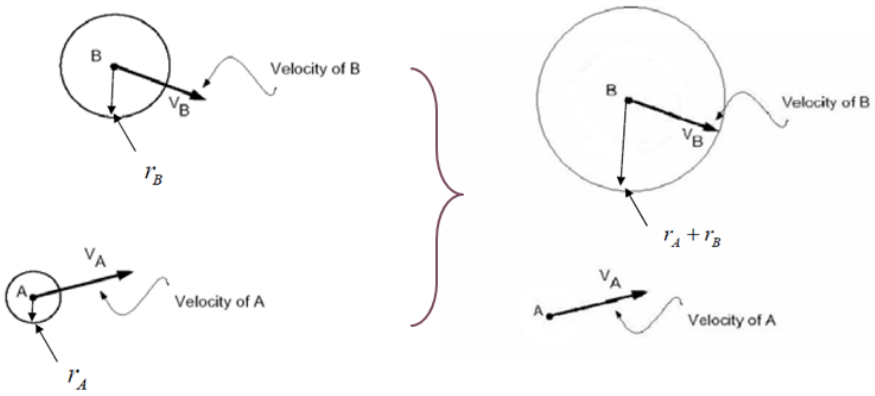

Figure 31: The ASV and a moving obstacle

Lets introduce the *collision cone*, $CC_{A,B}$, as a set of colliding relative velocities between $\hat{A}$ and $\hat{B}$

$$CC_{A,B} = \left\{ v_{A,B} \mid \lambda_{A,B} \cap \hat{B} \neq \emptyset \right\}, \tag{30}$$

with the relative velocity $v_{A,B}$ of $\hat{A}$ with respect to $\hat{B}$

$$v_{A,B} = v_A - v_B, \tag{31}$$

and $\lambda_{A,B}$ is the line of $v_{A,B}$.
This cone indicates a planer sector with apex in $\hat{A}$, bounded by the two tangents λ_f and λ_r from $\hat{A}$ to $\hat{B}$ as depict in Figure 32. Any relative velocity that lies between the two tangents λ_f and λ_r will cause a collision between A and B. Clearly, any relative velocity outside $CC_{A,B}$ is guaranteed to be collision-free, provided that obstacle $\hat{B}$ maintains its current shape and speed.

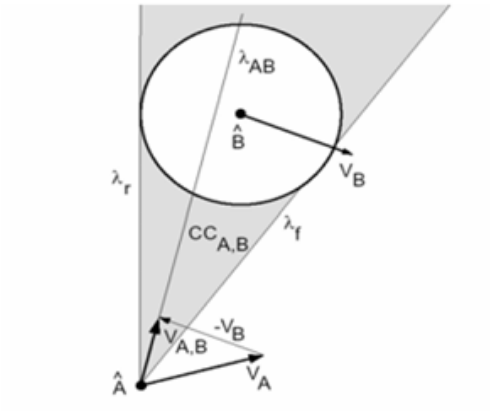

Figure 32: The collision cone $CC_{A,B}$

When considering multiple obstacles, it is useful to establish a condition on the *absolute velocity* of A.

This can be done by adding the velocity of B to each velocity in the cone $CC_{A,B}$ or equivalently by transforming the collision cone by v_B, as shown in Figure 33. The velocity obstacle VO can then be defined as

$$VO = CC_{A,b} \oplus v_B, \tag{32}$$

where $\oplus$ is the Minkowski vector sum operator [20].

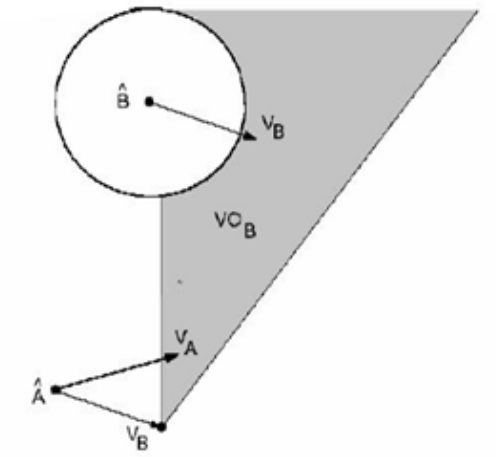

Figure 33: The velocity obstacle VO_B

The VO splits the absolute velocities of A into *avoiding* and *colliding* velocities. Selecting v_A outside of VO leads to a collision avoidance with B:

$$A(t) \cap B(t) = \emptyset \qquad \text{if} \quad v_A(t) \notin VO(t). \tag{33}$$

Velocities along the boundaries λ_f and λ_r result in A grazing B. Note that the VO of a stationary obstacle is identical to its relative velocity cone with $v_B = 0$.

Multiple obstacles can be avoided by considering the union of the individual velocity obstacles

$$VO = \bigcup_{i=1}^{n} VO_{B_i}, \tag{34}$$

where n is the number of obstacles.

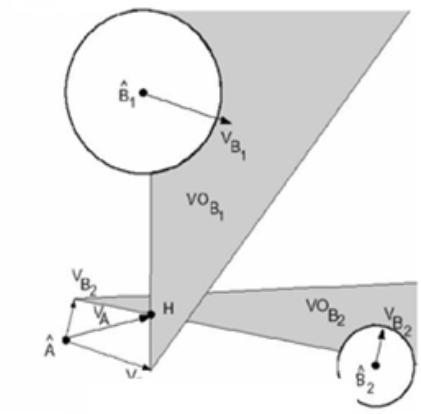

Figure 34: Velocity obstacles for multiple obstacles B_i

In the case of multiple obstacles the ones with imminent collision will be prioritized. A collision is called *imminent* if it occurs at some time $t < T_h$, where T_h is a suitable time horizon. Based on the system dynamics the time horizon T_h is computed by assuming that the ASV and the obstacle are moving at low speed in almost the same direction. To look at imminent collisions the VO set is modified by subtracting the VO_h set from it, defined as

$$VO_h = \left\{ v_A \mid v_A \in VO, \|v_{A,B}\| \leq \frac{d_m}{T_h} \right\}, \tag{35}$$

with d_m as the shortest relative distance between the ASV and the obstacle. This set specifies velocities that would result in collision, occurring beyond the time horizon.

Therefor this set is not relevant for the avoidance maneuver and can be removed from the VO set, as shown in the Figure below.

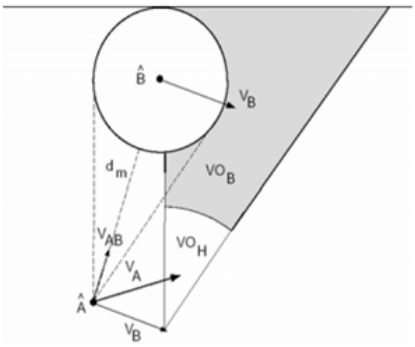

Figure 35: Velocity obstacles VO_B for a given time horizon

5.4 Avoidance Maneuver

Based on the assumption that the vessel's heading is simplified, obstacle detection and probable avoidance maneuvers are only computed during the phase where the vessel is not in a tack or jibe mode.
To avoid collisions with moving obstacles within a given time horizon an *avoidance maneuver* has to be introduced. It consists of a one-step change in the velocity vector of the ASV. First, the set of reachable velocities that account for the vessel's dynamics and actuator constraints will be discussed.

Reachable Avoidance Velocities

The velocities that can be reached by the vessel A at a given state over a given time interval Δt can be calculated by mapping the actuator constraints to acceleration constraints. A set of *feasible accelerations FA(t)* is defined as

$$FA(t) = \{\ddot{x} \mid \ddot{x} = f(x, \dot{x}, u), u \in U\}, \tag{36}$$

where x is the position vector, $f(x, \dot{x}, u)$ the vessel's dynamics, u the vector of the actuator efforts and U the set of admissible controls.
Thus the set of *reachable velocities* $RV(t + \Delta t)$, over the time interval Δt is defined as

$$RV(t + \Delta t) = \{v \mid v = v_A(t) \oplus \Delta t \cdot FA(t)\}, \tag{37}$$

Finally the set of *reachable avoidance velocities* RAV is defined as the difference between the reachable velocities and the velocity obstacle:

$$RAV(t + \Delta t) = RV(t + \Delta t) \ominus VO(t), \tag{38}$$

where $\ominus$ denotes the operation of set difference. By selecting any velocity in RAV a maneuver to avoid obstacle B can be computed.

Structure of the avoidance maneuvers

To choose the structure of the avoidance maneuver it is important to determine on which side of the obstacle each maneuver will take place. Therefor one identifies the front and rear sides of $\hat{B}$ by attaching a coordinate frame (X, Y) to the center of $\hat{B}$, oriented in a way that its X-axis coincides with the velocity v_B. The Y-axis partitions the boundary of $\hat{B}$ at the points Y_f and Y_r into the $front/rear$ semicircle $\delta B_f/\delta B_r$, as shown in Figure 36.

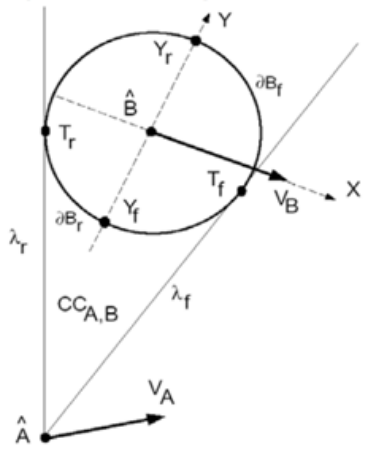

Figure 36: Grazing arcs in an avoidance maneuver

The following lemma [20] states that a maneuver along the tangents can only be created by the relative velocity lying on λ_f or λ_r. It also sais that the current tangency point of this maneuver is different from the tangency point T_f or T_r since they depend on the absolute velocity of A and B. In Figure 37 the position of the tangency point P at time t_0 is schematically shown, when the avoidance maneuver is generated, and at time t_1 when $\hat{A}$ reaches $\hat{B}$, assuming no rotations.

LEMMA1. Robot A is tangent to obstacle B at some point $P \in \delta B$ if and only if it follows a trajectory generated by v_A corresponding to $v_{A,B} \in \{\lambda_f, \lambda_r\}$. The tangency sets in δB consist of the shortest segment connecting $T_f = \lambda_f \cap \delta B$ to Y_f, and the shortest segment connecting $T_r = \lambda_r \cap \delta B$ to Y_r.

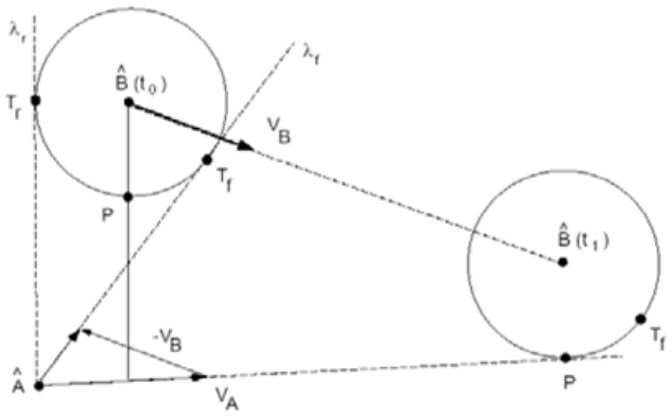

Figure 37: Trajectory tangent to B

This lemma is important in helping define the subsets of one type of avoidance maneuver in RAV.

In terms of safety reasons the ASV shall always select an avoidance maneuver passing the obstacle's rear boundaries. To gain more safety the radius of $\hat{B}$ can be increased. An other simplification is that the current speed of the ASV remains and can not be adjusted for such a maneuver. Therefore the set of *feasible accelerations* $FA(t)$ is zero and the vessel moves with its velocity at time t.

To avoid a collision the direction of the velocity vector of the ASV has to be changed. The main idea is to select the relative velocity lying on λ_r. To give the ASV a new heading, a new waypoint is computed. This is done with a geometric approach illustrated in Figure 38.

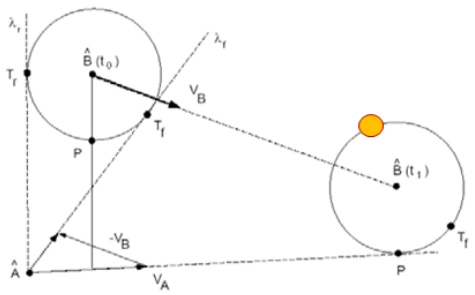

Figure 38: Computation of the new waypoint NP

The collision point will be calculated as well as the time t_s obstacle B needs to reach this point. With the given new relative velocity, the safe position NP (orange circle) of the vessel A, reachable within the same time t_s, can be computed. The new waypoint NP will then be addressed to the navigator and a collision is avoided.

6 Conclusion

This thesis focuses on the simulation and the controllability of the complex sail boat system. With a simulation environment introduced in chapter 3 the vessel's behavior can be represented and analyzed. Based on a validation of the kinematic model from chapter 2.2 a similar performance to the real system is provided. Therefor the simulator can be used as an efficient platform helping to optimize the software system. Extensive outdoor tests with the real system as well as time and cost consuming operations can be avoided. Another benefit of the simulator is that environmental conditions can be set arbitrarily, such that performance and functions can be systematically verified. Additionally it helps to simulate a wide range of typical failures of the sensors which are difficult or undesirable to simulate physically.
With the detection of an unstable control section a new rudder controller has been successfully established. Figure 39 depicts a trajectory generated by the new controller.

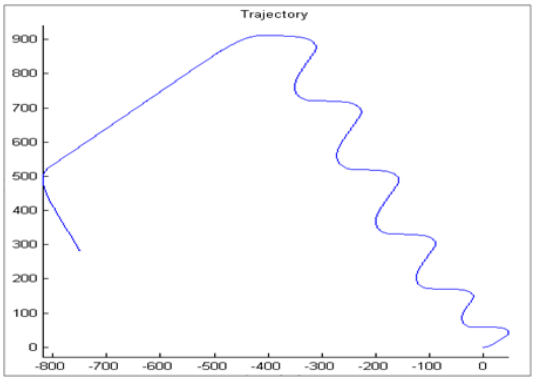

Figure 39: Trajectory of the new controller

It can be seen that the controller reduces the response to reach the desired rotation speed of the vessel. Therefor a more stable behavior is guaranteed but the maneuverability is decreased. This leads to a bigger curve radius the vessel has to sail when turning.

In chapter 5 a method for planing the motion of the vessel in a time-varying environment, the Velocity Obstacle approach, has been presented. It is mainly different from currently available obstacle avoidance algorithms, since it simultaneously generates the path and velocity profile that avoid all static and dynamic obstacles and satisfy the sail boat constraints.
For every obstacle the method computes its corresponding velocity obstacle, which is the set of colliding velocities between the vessel and the obstacle. Then, by subtracting it from the reachable velocities of the vessel, a set of reachable avoidance velocities is formed. It consists of all the velocities which guarantee the obstacle avoidance. The main advantage of this approach include the efficient geometric representation of maneuvers avoiding any number of static and moving obstacles.
The computed solutions of this method are conservative, since each single maneuver avoids all obstacles, irrespectively of their expected collision time. They possibly exclude trajectories which, although feasible, violate the velocity obstacle in some interval.

To test the optimized control system on the real system, first a Hardware-in-the-Loop test is done. In the first stage a computer with the simulator was connected to the sail boat. The communication between the simulator, the on-board computer and the actuators worked and environmental based variables from the simulator could be sent to the on-board computer. It will be explained in the Outlook how the future work with the hardware system can be done to guarantee safe sailing in the water.

7 Outlook

There are several possibilities to extend the work presented in this thesis. With the main focus on autonomy and safe sailing, research work in a relevant field can lead to a reliable and fully autonomous sail boat system:

3D Rudder Control

The optimized rudder controller is based on a kinematic model in the 2D plane. Considering that the boat rotates around its x-axis it may happen that one of the two rudders is above the water surface. In this case it is only one rudder acting on the vessel's rotation speed. The idea is to find a dependency of the rudder area in the water and the angular position of the vessel around its x-axis and to implement this relation into the rudder controller. A possible relation could be a linear dependency

$$A_r = A_r \cdot (1 - \frac{\beta}{2 \cdot \beta_{max}}) \tag{39}$$

where A_r is the area of both rudders, β the current angular position of the vessel around its x-axis and β_{max} the maximum angular position of the vessel around its x-axis where one of the rudder is completely out of the water. In this case the behavior of an inclined rudder has to be identified first.

Dynamic Obstacle Avoidance

Based on the presented method for dynamic obstacle avoidance by using velocity obstacles, the feasible accelerations of the boat systems, chapter 5.3, can be analyzed more detailed. At the moment a one-step change in the vessel's velocity direction is used to avoid obstacles. For a sailing expert it is also possible to decelerate the speed by turning the sail to the wind. Therefor the use of the sail acting in the wind must be established and formulated as a set of feasible accelerations. Implemented in the avoiding algorithm it would then be possible not only to change the velocity direction, but also to change the velocity to guarantee collision free maneuvers.

Error Recovery Software

The detection of sensor faults has to be extended. A model of a fault-free behavior must be established. With this model a prediction can be estimated and can then be compared to its observation pair. This comparison could be realized by generating a residual R_s as a function of the squared difference between real (c_i) and estimated ($\hat{c}_i$) sensor outputs:

$$R_s = \sum_{i=1}^{n} m_i(c_i - \hat{c}_i) \tag{40}$$

where m_i are weighting coefficients that have to be determined for each failure based on experience and experimentations.

Hardware-in-the-Loop

To gain more information about the system's performance, several Hardware-in-the-Loop simulations must be executed. Starting with a long-term journey the properties of the software system behavior can be analyzed and the error recovery software can be improved if necessary. Later on, a one-day outdoor test must be conducted to parse the control behavior in the real-world. After obtaining satisfying results the sail boat can be prepared for an autonomous cross over the Atlantic Ocean.

Bibliography

References

[1] Karl Johan Astrom, Claes G. Kallstrom. "The Identification of Linear Ship Steering Dynamics using Maximum Likelihood Parameter Estimation". Statens Skeppsprovningsanstalt, Goeteborg (SE), 1975.

[2] Claes G. Kallstrom. "Identification and Adaptive Control applied to Ship Steering". Marintekniska Institutet SSPA, Goeteborg (SE), 1982.

[3] Tristan Perez. "Ship Motion Control". Springer-Verlag, London (GB), 2005.

[4] Anthony F. Molland, Stephen R. Turnock. "Marine Rudders and Control Surfaces". Elsevier Ltd., Oxford (US), 2007.

[5] Roland Siegwart, Illah R. Nourbakhs. "Introduction to Autonomous Mobile Robots". MIT Press, Camridge (US), 2004.

[6] Hendrik Erckens, Gion-Andri Büsser, Dr. Cédric Pradalier, Prof. Dr. R. Siegwartand. "Navigation Strategy and Trajectory Following Controller for an Autonomous Sailing Vessel", *IEEE Trans. RAM*, Zurich (CH), 2009.

[7] P.Corke, P.Sikka, J.Roberts, E.Duff. "Ddx: A distributed software architecture for robotic systems", in *Proc. Australian Conf. Robotics and Automation*, Canberra, 2004.

[8] Y.Briere. "IBOAT: An autonomous robot for long-term offshore operation", in *Electrotechnical Conference*, Melecon, 2008. *The 14th IEEE Mediterranean, 2008, pp. 323-329*

[9] Tor A. Johansen, Thor Fossen, Bjornar Vik. "Hardware-in-the-loop Testing of DP systems", *Dynamic Positioning Conference*, Norway, 2005.

[10] Roland Stelzer, Tobias Pröll. "Autonomous sailboat navigation for short course racing", *Robotics and Autonomous Systems*, Leicester (UK), October 2007.

[11] Sang-Min Lee, Kyung-Yub Kwon. "A Fuzzy Logic for Autonomous Navigation of Marine vehicles Satisfying COLREG Guidelines", *International Journal of Control, Automation and Systems*, Vol. 2, No. 2, June 2004.

[12] Jaime Abril, Jaime Salom. "Fuzzy Control of a Sailboat", *International Journal of Approximate Reasoning*, New York (USA), 1997.

[13] Philip J. Sterne. "Reinforcement Sailing", University of Edinburgh, Scotland, 2004.

[14] November 2009, http://www.microtransat.org.

[15] Dezember 2009, http://www.gnu.org/software/gsl.

[16] Paolo Fiorini, Zvi Shiller. "Motion Planning in Dynamic Environments Using Velocity Obstacles", *The International Journal of Robotics Research*, Vol. 17, Sage, 1998.

[17] Yang Zeng, Hayama Imazu. "Algorithm for Cooperative Collision Avoidance through Communication with Autamtic Idetification System", *International Conderence on Computational Intelligence for Modeling, Control and Automation*, 2005.

[18] Chiara Fulgenzi, Anne Spalanzani, Christian Laugier. "Combining Probabilistic Velocity Obstacles and Occupancy Grid for safe Navigation in dynamic environments", *ICRA Workshop: Planning, Perception and Navigation for Intelligent Vehicles*, 2007.

[19] Paolo Fiorini, Zvi Shiller. "Time Optimal Trajectory Planning in Dynamic Environments", *IEEE International Conference on Robotics and Automation*, Minneapolis, Minnesota (USA), April 1996.

[20] Zvi Shiller, Frederic Large, Sepanta Sekhavat. "Motion Planning in Dynamic Environments: Obstacles Moving Along Arbitrary Trajectories", *IEEE International Conference on Robotics and Automation*, Seoul, Korea, Mai 2001.

[21] Jur van den Berg, Ming Lin, Dinesh Manocha. "Reciprocal Velocity Obstacles for Real-Time Multi-Agent Navigation", *ARO Contracts*, North Carolina (USA), 2001.

[22] Parris K. Egbert, Scott H. Winkler. "Collision-Free Object Movement Using Vector Fields", *IEEE International Conference on Robotics and Automation*, Brigham Young University , 1996.

[23] Sven Baselau, Felix Hahne, Klaus Ambrosi. "Shortest-Path Algorithms and Dynamic Cost Changes", *Institut für Betriebswirtschaft und Wirtschaftinformatik*, Universität Hildesheim, 2002.

[24] Nicholas Villamagna, Peter A. Crossley. "A Symmetrical Component-Based GPS Signal Failure-Detection Algorithm for use in Feeder Current Differential Protection", *IEEE Transactions on Power Delivery*, Vol. 23, No. 4, October 2008.

[25] Hua-Zhi Hsu, Neil A. Witt, John B. Hooper, anne P. McDermott. "The AIS-Assisted Collision Avoidance", *The Journal of Navigation*, Vol. 62, pp. 657-670, United Kingdom, 2009.

[26] Rongfu Luo, Manish Misra, Tyler Soderstrom, David M. Himmelblau. "Sensor Fault Detection via Multiscale Analysis of Prediction Model Residuals", *American Chemical Society*, 2003.

[27] Wolfgang Reinelt, Andrea Garulli, Lennart Ljung. "Comparing different approaches to model error modeling in robust identification", *Automatica 38*, 2002.

[28] Guillermo Heredia, Anibal Ollero, Rajesh Mahtani, Manuel Bejar. "Detection of Sensor Faults in Autonomous Helicopters", *IEEE International Conference on Robotics and Automation*, Barcelona (ES), April 2005.