

A

comment



Stellenbosch
UNIVERSITY
IYUNIVESITHI
UNIVERSITEIT

forward together
sonke siya phambili
saam vorentoe

Control and Development of a Sailboat for Autonomous Racing

Johann Ruben van Tonder
22569596

Thesis presented in partial fulfilment of the requirements for the degree of Master of Engineering (Electronic) in the Faculty of Engineering at Stellenbosch University

Supervisor: Prof. Jaco Versfeld & Prof. Japie Engelbrecht
Department of Electrical and Electronic Engineering

September 2024

Acknowledgements

I would like to sincerely thank the following people for assisting me in the completion of my project:



UNIVERSITEIT•STELLENBOSCH•UNIVERSITY
jou kennisvennoot • your knowledge partner

Plagiaatverklaring / Plagiarism Declaration

1. Plagiaat is die oorneem en gebruik van die idees, materiaal en ander intellektuele eiendom van ander persone asof dit jou eie werk is.

Plagiarism is the use of ideas, material and other intellectual property of another's work and to present it as my own.

2. Ek erken dat die pleeg van plagiaat 'n strafbare oortreding is aangesien dit 'n vorm van diefstal is.

I agree that plagiarism is a punishable offence because it constitutes theft.

3. Ek verstaan ook dat direkte vertalings plagiaat is.

I also understand that direct translations are plagiarism.

4. Dienooreenkomsdig is alle aanhalings en bydraes vanuit enige bron (ingesluit die internet) volledig verwys (erken). Ek erken dat die woordelikse aanhaal van teks sonder aanhalingstekens (selfs al word die bron volledig erken) plagiaat is.

Accordingly all quotations and contributions from any source whatsoever (including the internet) have been cited fully. I understand that the reproduction of text without quotation marks (even when the source is cited) is plagiarism

5. Ek verklaar dat die werk in hierdie skryfstuk vervat, behalwe waar anders aangedui, my eie oorspronklike werk is en dat ek dit nie vantevore in die geheel of gedeeltelik ingehandig het vir bepunting in hierdie module/werkstuk of 'n ander module/werkstuk nie.

I declare that the work contained in this assignment, except where otherwise stated, is my original work and that I have not previously (in its entirety or in part) submitted it for grading in this module/assignment or another module/assignment.

Studentenommer / Student number	 Handtekening / Signature
Voorletters en van / Initials and surname	Datum / Date

Abstract

English

Afrikaans

Contents

Declaration	ii
Abstract	iii
List of Figures	viii
List of Tables	xi
Nomenclature	xii
1. Introduction	1
1.1. Background	1
1.2. Problem Statement	1
1.3. Summary of Work	1
1.4. Scope	1
1.5. Format of Report	1
2. Literature Review	2
2.1. Related work on USV's	2
2.1.1. South Africa Sailboat	2
2.1.2. Italy Sailboat	3
2.1.3. Norwegian Sailboat	4
2.1.4. Sweden Sailboat	5
2.2. Control techniques for USV's	6
2.2.1. Rudder Control	6
2.2.2. Sail Control	9
2.3. Platform development of USV	9
2.4. Usages of USV	11
2.5. Ardupilot	12
3. System Overview	13
3.1. Physical System	13
3.2. Marine Hardware	14
3.2.1. Autopilot Controller	15
3.2.2. GPS and Compass Sensor	16

3.2.3. Telemetry System	17
3.2.4. Wind Vane	17
3.3. Stationary Base-Station	18
3.3.1. Ground Control Software	18
3.4. Simulation Software	21
3.4.1. Matlab Simulator	21
3.4.2. ArduPilot Simulation	21
4. Ocean Vessel Dynamic Model	23
4.1. Reference Frames and Notations	23
4.1.1. Ocean Vessel Notation	24
4.1.2. Earth-Centered Reference Frames	24
4.1.3. Geographic Reference Frame	25
4.2. Rigid-Body Equations of Motion	26
4.2.1. Kinetics	26
4.2.2. Kinematics	27
4.3. Hydrodynamic and Hydrostatic Forces and Moments	27
4.3.1. Restoring Forces and Moments	29
4.4. Environmental Disturbances	31
4.4.1. Current-induced Forces and Moments	31
4.4.2. Wave-induced Forces and Moments	32
4.4.3. Wind-induced Forces and Moments	32
4.5. Simplified Maneuvering Model including Roll(4 DOF)	34
4.6. Modelling of a Sail, Keel and Rudder	35
4.6.1. Rudder Theory	35
4.6.2. Sail Theory	37
4.7. Linearizations and Simplifications	45
5. Sailing Control System Development	46
5.1. Sailing Control System Overview	46
5.2. Autopilot Design	47
5.2.1. Ship Steering Dynamics Model Reduction	47
5.2.2. First Order Nomoto Model	49
5.2.3. Stabilization	52
5.2.4. Acceleration Feedback	58
5.2.5. Proposed other Control Techniques	58
5.3. Sail Controller Design	59
5.4. Guidance and Waypoint Navigation	62
5.4.1. Guidance System	62
5.4.2. Straight path Generation based on Waypoints	65

5.4.3. Line-of-Sight (LOS) Steering Law	66
5.4.4. Path-Following Controllers	69
5.4.5. Circle of Acceptance for Surface Vessel	71
5.5. Sailboat Maneuvering	72
5.5.1. The Three forms of Sailboat Maneuvering	72
5.5.2. Finite State Machine of Sailboat Maneuvering	74
5.5.3. Heading Controller	74
5.5.4. Tacking Controller	76
6. Ardupilot	78
6.1. Architecture	78
6.2. Rover Architecture	79
6.3. Sailboat Framework	83
6.3.1. The Shortcomings of Sailboat Framework	84
6.4. Ardupilots Control Strategies	85
6.4.1. Sail Controller	85
6.4.2. Rudder Controller	85
6.4.3. Roll Controller	86
6.4.4. Tacking	86
6.4.5. Waypoint Navigation	87
7. Simulations	88
7.1. RSS Simulator	88
7.1.1. Additions in RSS	89
8. Results	90
9. Conclusion	91
Bibliography	92
A. Additional Modelling Information	95
A.1. Notation and Vector Definitions	95
A.2. Modeling Equations	96
A.2.1. Rigid-Body System Inertia Matrix M_{RB}	96
A.2.2. Rigid-Body Coriolis and Centripetal Matrix C_{RB}	96
A.3. Ocean Vessel Parameters	98
A.4. Heading Controller	98
A.5. Matlab Code	99
A.5.1. Sail Controller	99
A.5.2. Heading Controller	99

A.5.3. Waypoint Controller	101
A.5.4. Waypoint Construction	103
A.6. Sailboat Specifications	103
A.7. Pixhawk 6C Technical Specifications	104
A.8. M8N GPS Technical and Features Specifications	107
A.9. Ardupilots Sailboat Configuration Parameters	109
A.10. Copter Modes Architecture	111
A.11. Creating Custom Log Messages	113

List of Figures

2.1.	UCT Sailboat	2
2.2.	Variable draft sailing spar	3
2.3.	Aeolus, sailing boat of ETH Zurich	4
2.4.	Sailboat used for modeling	5
2.5.	Sailboat provided by University of Hong Kong	5
2.6.	Reference heading α^* , obtained angle α and rudder input δ	7
2.7.	Raw and average wind direction σ raw and σ avg, heading ψ and course over ground X	7
2.8.	Rule based controller for sail	9
2.9.	Platform developed by UCT sailboat	10
2.10.	GUI developed for sailboat monitoring	11
2.11.	Saildrone's USV	11
2.12.	Routes of the USV's in Arctic Mission	12
3.1.	Dragonflite 95 RC model sailboat	13
3.2.	Block diagram overview of physical system	14
3.3.	Block diagram of marine hardware	15
3.4.	Pixhawk 6C	16
3.5.	M8N GPS	16
3.6.	Telemetry radio	17
3.7.	High Level Communication	17
3.8.	Caption	18
3.9.	Block diagram of base station	18
3.10.	Mission Planner GUI	19
3.11.	An example of a mission consisting of waypoints and a fence	20
3.12.	Overview of Matlab simulator	21
3.13.	Architecture of Ardupilots SITL	22
4.1.	The Earth-centered Earth-fixed (ECEF) frame and an Earth-centered inertial (ECI) frame	23
4.2.	Motion variables for an ocean vessel	24
4.3.	Transverse metacentric stability	29
4.4.	Ocean vessel's heading angle	32
4.5.	Wind angle on vessel	33

4.6. Terminology of a lifting foil	35
4.7. Pressure around the rudder	35
4.8. Lift and drag force on rudder	36
4.9. Definition of Rudder Angles	37
4.10. Naming convention of a sail	38
4.11. Terminology of a sail wing and shape	38
4.12. Windward and leeward definitions	39
4.13. Flow field without circulation	39
4.14. Superposition of circulation and non-circulation solution to give lift	40
4.15. Lift and drag forces due to wind on sail wing	41
4.16. Lift force component represented as a power curve	44
5.1. Block diagram overview of SCS	46
5.2. Sway-yaw-roll motion coordinate system	47
5.3. First Order bode Plot	51
5.4. Second Order Bode Plot	52
5.5. Fourth Order Bode Plot	52
5.6. Straight Line stability	55
5.7. Directional Stability(critically damped)	56
5.8. Directional Stability(under damped)	56
5.9. Positional Motion Stability	57
5.10. PID rudder controller	57
5.11. Rudder reference angle	59
5.12. Sail Reference Angle	59
5.13. Optimal sail angle for varying wind angle	61
5.14. Dimensionless sail force	62
5.15. Ground track between source waypoint and destination waypoint	63
5.16. Cross-track error and in-track distance along track	63
5.17. Guidance Axis System	64
5.18. Straight lines and circular arc segments for waypoint guidance	66
5.19. LOS guidance where the desired course angle x_d is chosen to point toward the LOS intersection point (x_{los}, y_{los})	67
5.20. Circle of acceptance with constant radius R	68
5.21. LOS guidance principle where the side slip angle B can be applied and compensated for by using integral action	70
5.22. Definition of vector \bar{u} and \bar{v}	70
5.23. Sail Zone	72
5.24. Tacking	73
5.25. Jibbing	73

5.26. Normal Sailing	73
5.27. Finite-State Machine for Sailboat Maneuvering	74
5.28. Finite-State Machine for Sailboat Maneuvering	75
5.29. Indirect tack calculations	77
6.1. High Level Architecture of Ardupilot Software	79
6.2. Rover architecture	80
6.3. Rover Controllers	81
6.4. Rover navigation overview	81
6.5. Formulas for L1 controllers	82
6.6. Code Execution Flow for Sailboat in auto mode	84
6.7. ArduPilots Rudder Controller	85
6.8. Tacking	86
6.9. ArduPilots Waypoint Navigation	87
7.1. High level block diagram overview	89
A.1. Pixhawk 6C Dimensions	106
A.2. Pixhawk 6C Pinout Diagram	107
A.3. M8N GPS pinout diagram	108
A.4. Manual Flight Mode	111
A.5. Autonomous Flight Mode	112

List of Tables

4.1. SNAME Notation for Ocean Vessels	24
5.1. Definitions of Actions in State Machine 5.27	74
5.2. Definitions of Actions in State Machine 5.28	75
A.1. Rigid body motion vectors	95
A.2. Sail Angle Look-up Table	97
A.3. Parameters of Container Vessel	98
A.4. Specifications of Dragonflite 95 Sailboat	103
A.5. Sailboat Configuration Parameters	109
A.5. Sailboat Configuration Parameters	110

Nomenclature

Ocean Vessel Dynamics

X, Y, Z	Coordinates of force vector decomposed in the body-fixed frame(surge, sway and heave forces)
K, M, N	Coordinates of moment vector decomposed in the body-fixed frame(roll, pitch and yaw moment)
u, v, w	Coordinates of linear velocity vector decomposed in the body-fixed frame(surge, sway and heave velocities)
p, q, r	Coordinates of angular velocity vector decomposed in the body-fixed frame(roll, pitch and yaw angular velocities)
x, y, z	Coordinates of position vector decomposed in the body-fixed frame(surge, sway and heave positions)
ϕ, θ, ψ	Coordinates of Euler angle vector decomposed in the body-fixed frame(roll, pitch and yaw Euler angles)

Guidance and Navigation

x_i, y_i, ψ_i	Coordinate system for guidance control known as the <i>pose</i>
$\psi_{heading}$	Track heading
L_{track}	Track length
(E_{src}, N_{src})	Source Waypoint
(E_{dest}, N_{dest})	Destination Waypoint

Acronyms and abbreviations

SNAME	Society of Naval Architects and Engineers
CG	Centre of gravity of a vessel
CB	Centre of buoyancy of a vessel
AoA	Angle of attack
Re	Reynolds number
USV	Unmanned Surface Vehicle
AWA	Apparent Wind Angle
AWV	Apparent Wind Velocity
RC	Remote Control
GPS	Global Positioning System
MAVLink	Micro Air Vehicle Link
RTL	Return-To-Launch
GCS	Ground Control System
HITL	Hardware-In-The-Loop
SITL	Software-In-The-Loop
HUD	Heads-Up Display
HAL	Hardware Abstraction Layer
RSS	Ruben's Sailboat Simulator
SCS	Sailing Control System
ECI	Earth-Centered Inertial
ECEF	Earth-Centered Earth-Fixed

Sailing Terminology

Bow	Front of the sailboat
Stern	Rear of the sailboat
Luff	Leading edge of the sail
Leech	Trailing edge of the sail
Foot	Bottom edge of the sail
Boom	Attached point of the foot to the sailboat
Clew	Attachment point of the leech
Tack	Attachment point of the luff
Chord	Straight line between leading and trailing edge
Camber	Perpendicular distance from the chord line to the foil
Draft	Position of the maximum camber along the chord line
Entry	Angle of the leading edge to the chord line
Exit	Angle of the trailing edge to the chord line
AoA	Angle between oncoming flow and the chord line
Twist	Angle between the chord line and the sailboat's centre line

Chapter 1

Introduction

1.1. Background

1.2. Problem Statement

The problem is that of modeling and controlling a fixed wing unmanned surface vehicle(USV), or in layman's terms a sailboat. Control strategies and techniques are applied to find the best way of controlling a USV on a line trajectory and performing speed control.

1.3. Summary of Work

The code of the modified repositories is available on my GitHub account [1].

1.4. Scope

1.5. Format of Report

Chapter 2

Literature Review

In recent years the development of robotics has enabled a new era of exploration. This exploration is made possible through USV's, more commonly known as sailboats.

Sailboats can be used to explore areas of the ocean previously unexplored. The sailboat are equipped with an array of sensors that collect data on important ocean factors and the ecosystem inside. This chapter will review the usage of sailboats to collect data and exploration usages. Also it will look into the control techniques used in sailboats and how a sailboat is controlled to stay on the designated course.

2.1. Related work on USV's

2.1.1. South Africa Sailboat



Figure 2.1: UCT Sailboat

The first sailboat used for review is close to home. The sailboat was developed by the University of Cape Town(UCT) [2], the sailboat is illustrated in Figure 2.1. The study focussed on modeling and simulation of a sailboat. No sort of control techniques was

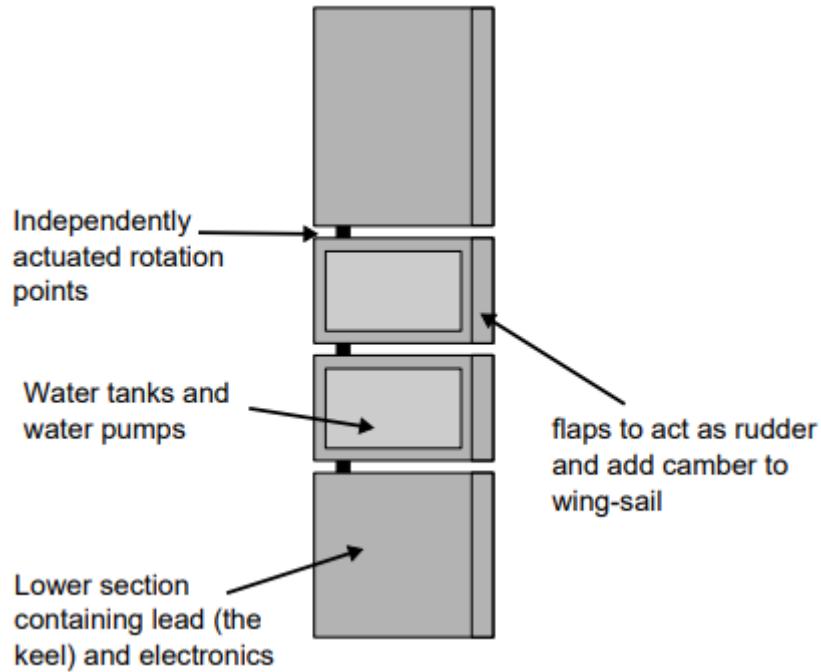


Figure 2.2: Variable draft sailing spar

developed. The modelling of the sailboat was done in the three and four degrees of freedom, which is the most popular degrees of freedom when considering sailboats. The four degrees of freedom are surge, sway, yaw and roll. The field tests collected valuable data which agreed with the limitations mentioned in the study. The study used a fixed wing sail but also proposed a novel sail trimming on a fixed wing sail. The platform that was developed to monitor the sailboat performed well in field test but there was one problem with the wind sensor not giving reliable data. The field tests were executed on dams near UCT.

2.1.2. Italy Sailboat

Now lets move over to Europe, more specifically Italy and the University of Pisa with Aeolus [3]. Aeolus is a small model sized sailing boat, provided by ETH Zurich, fitted with a wind sensor. The sailboat makes use of a fabric sail unlike the sailboat illustrated in Figure 2.1. Aeolus makes use of a main sail and a jib. The sail controller is only used to adjust the main sail angle.



Figure 2.3: Aeolus, sailing boat of ETH Zurich

The study focussed on developing navigation and control strategies for an autonomous sailing model boat. In the study different rudder controllers were implemented to perform the maneuvering technique called tacking and performed field test in the lake Zurich, Switzerland. The study successfully implemented a turning dynamic, real time data collecting and filtering, sailing upwind and executing tack maneuvers.

2.1.3. Norwegian Sailboat

Not far from Zurich lies the ancestors of the vikings who mastered the arts of sailing. The Norwegian model [4] was based on the sailboat illustrated in Figure 2.4. The thesis focused on modeling, simulation and control of a sailboat. The study first developed a mathematical model in four degrees of freedom. From this mathematical model the rudder and sail controller was designed. New strategies for a course control and a technique to reduce roll motion was also developed. A path following strategy was created and state estimation through sensors was created. All of the above mentioned controllers and strategies was tested and achieved desirable results.



Figure 2.4: Sailboat used for modeling

2.1.4. Sweden Sailboat



Figure 2.5: Sailboat provided by University of Hong Kong

The study focused on developing a mathematical model with simulations of the control strategies. The control was designed for optimal trajectory approach. The tests were confined to indoor tests where fans were used to generate wind. The fans caused turbulence which impacted the tests. The modeling did not take into consideration the roll, pitch and heave and the control strategies still performed accurately. The controllers implemented were a PD controller and the force polar diagram for the sail angle.

2.2. Control techniques for USV's

The control of a USV is a problem of steering the vehicle in the appropriate direction and controlling the sail angle. The two common sails used are either a tradition fabric sail [5] or a fixed wing sail [6]. The steering of a vehicle is made possible through controlling the rudder. A USV is known as a underactuated vessel, which is when a vehicle has uncontrollable states. The only controllable states are the yaw rate and surge speed, which is defined in Section ??.

2.2.1. Rudder Control

The sailboat made use of a basic rudder control, which did not achieve desirable results. The controller used to control the rudder is a lead controller***expand***, although their was reported that it performed badly due to amplifying the high frequency noise. The study concluded that their is still significant amount of work required on the rudder control.

Aeolus makes use of two rudder controllers either using the one or the other depending on the circumstance. When the error signal e , which is the difference between the desired heading and the actual heading, is small the controller is defined as

$$\delta = k_p e \quad (2.1)$$

this is a simple Proportional controller for the rudder at small error signals. The second controller is used in the case of the error signal being large when implementing a non-linear controller. The non-linear controller defines a nonlinear gain $k(e)$ as

$$k(e) = \frac{k_p}{1 + c_p|e|} \quad (2.2)$$

The constant c_p is used to tune the control action when the error is large. This enables the sailboat to perform special maneuvering called tacking. The performance of the rudder controller is illustrated in Figure 2.6 and 2.7. The rudder controller performed reasonably accurate apart from suffering from drift which causes a difference in the heading ψ and course over ground X .

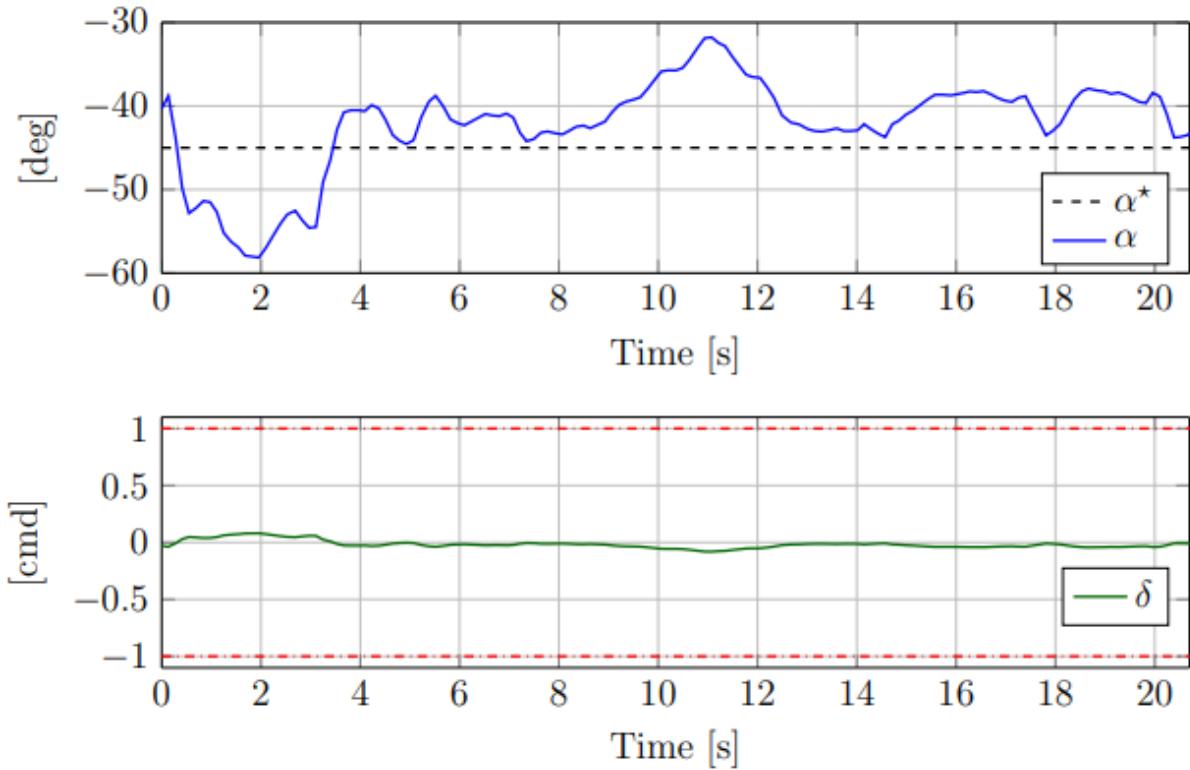


Figure 2.6: Reference heading α^* , obtained angle α and rudder input δ

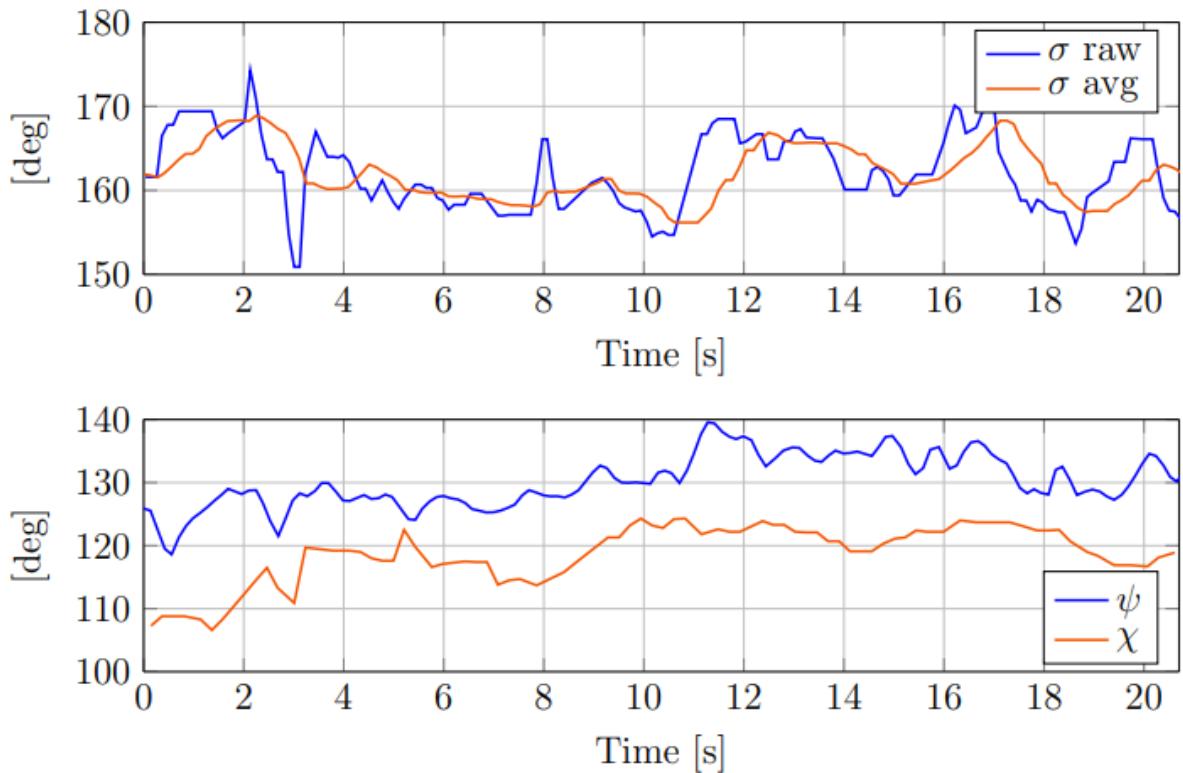


Figure 2.7: Raw and average wind direction σ raw and σ avg, heading ψ and course over ground X

The rudder control of the Norwegian sailboat is by far the most complex discussed thus

far, the controller is using control Lyapunov functions and a recursive design approach. Where the new state variables are,

$$z_1 = \psi + \beta_b + X_d \quad (2.3)$$

and

$$z_2 = r - \alpha \quad (2.4)$$

where X_d is the desired heading, z_1 is the error from the desired course angle, ψ is the actual heading and β_b is the necessary drift angle such that the desired course angle is achieved. β_b is a bias, the drift and course angle is defined as

$$\beta = \arctan2(v\cos(\phi), u) \quad (2.5)$$

$$X = \psi + \beta \quad (2.6)$$

The goal of the controller is to make the Lyapunov Function stable,

$$\dot{V}_1 = -K_p z_1^2 + z_1 z_2 \quad (2.7)$$

Solving the controller action u results in the state z_1 to due the following procedure,

$$z_1 = \text{mod}(z_1 + \pi, 2\pi) - \pi \quad (2.8)$$

where mod is the modulo operation. This ensures that the boat always turns in the direction that has the smallest error. If one calculates z_1 one will always end up tacking up-wind an jibing down-wind. The correction term β_b is used to compensate for the drift angle, by using a lookup-table that stores the necessary drift angles to keep a desired course. A simpler method was introduced in this study, where z_1 state of a traditional heading controller is equal to

$$z_1 = \psi - \psi_d = \psi - (X_d - \beta_b) = \psi + \beta_d - X_d \quad (2.9)$$

The study found the following solution for β_b as follows

$$\beta_b = -\alpha_k \cos(\phi) \quad (2.10)$$

where

$$\alpha_k = -\frac{\beta}{\cos(\phi)} \quad (2.11)$$

The lift the rudder can create is limited which limits the maximum moment can create in yaw. The controller does not take this into account and therefore to smooth the rudder

response a second order low-pass filter was added. The transfer function is shown below,

$$F(s) = \frac{\omega^2}{s^2 + 2\omega + \omega^2} \quad (2.12)$$

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

2.2.2. Sail Control

The sailboat of the UCT sailboat was controlled by setting the sail angle of attack to a predetermined angle depending on the apparent wind angle. The apparent wind angle was divided into three regions $-90 < \beta < 90$, $-160 < \beta < 160$ and $-180 < \beta < 180$. The hardware on the ETH Zurich sailboat platform did not allow for feedback measurements of the position of the sail and therefore a simple open loop controller was implemented. The controller was based on a rule, the closer Aeolus is sailing opposite the wind direction, that is closer to 0° , the more the sail is closed. The rule based controller is shown in Figure 2.8. The parameters x_1 and x_2 can be changed at run time and the higher the sail command μ , the more the sail is closed.

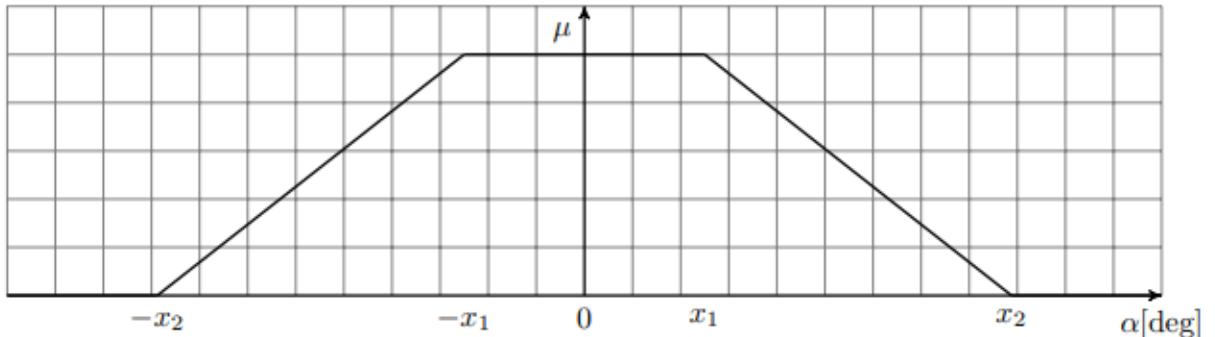


Figure 2.8: Rule based controller for sail

To obtain the optimal sail angle to give the highest forward acceleration for a given wind direction, making λ_d , which is the sail angle, a function of β_{ws} , $\lambda_d(\beta_{ws})$. This was achieved by defining a function independent of wind speed and measuring the force it generates. The optimal sail angle was stored in a look-up table, taking into account the restriction of possible sailing angles depending on the wind direction.

2.3. Platform development of USV

The platform required for sailboat control is quite simple. It requires a rudder angle feedback, compass or gyro, wind sensor, GPS sensor and the methods for controlling the sail and rudder. Accompanying these sensors onboard an off board station is required to monitor the sailboat's state. The difficulty in designing and implementing the platform is

the limitation of space available on the model sized sailboats. The goal of these small scale sailboats are to test control strategies for navigation and route planning. It is very seldom that you would add any other sensors apart from cameras used for obstacle avoidance.

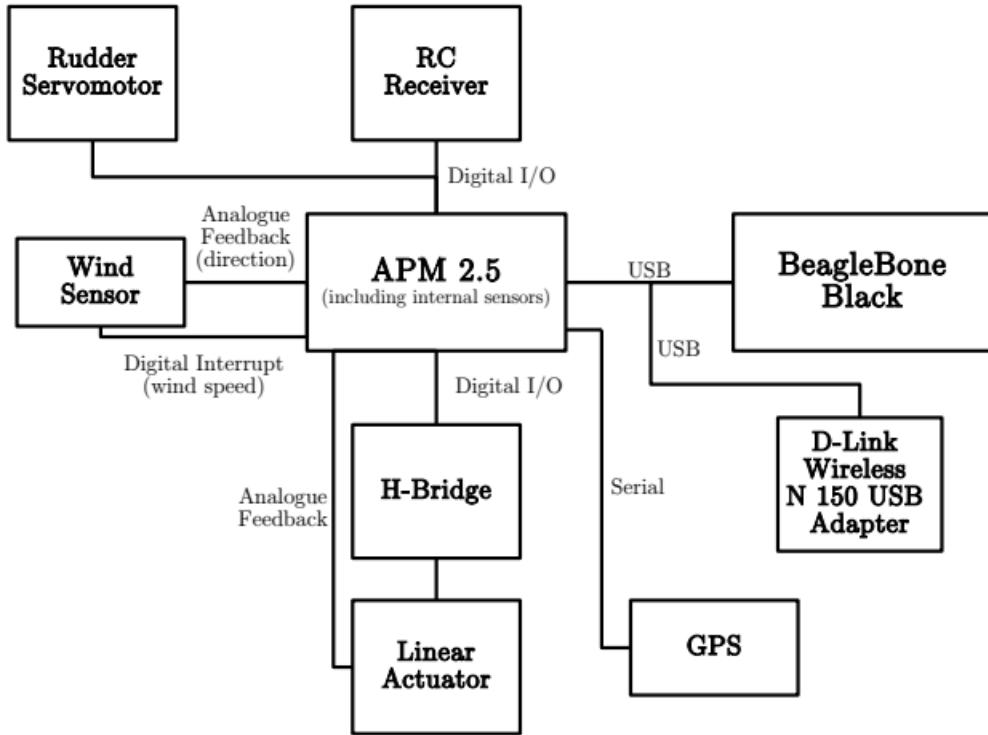


Figure 2.9: Platform developed by UCT sailboat

As illustrated in Figure 2.9 the rudder is controlled with a servomotor and the sail with a linear actuator. It is quite common to make use of either control methods when it comes to controlling the sail or rudder. One problem noted in [2], is the inaccurate data from the wind sensor and this will be taken into account when designing the platform. The next important part of the platform is the base station used for communicating with the sailboat. One such base station GUI developed by [3]. The base station, as previously mentioned, monitors the important states of the sailboat, as the heading angle, rudder and sail command. The base station is also used to send data and commands to the sail boat when required.

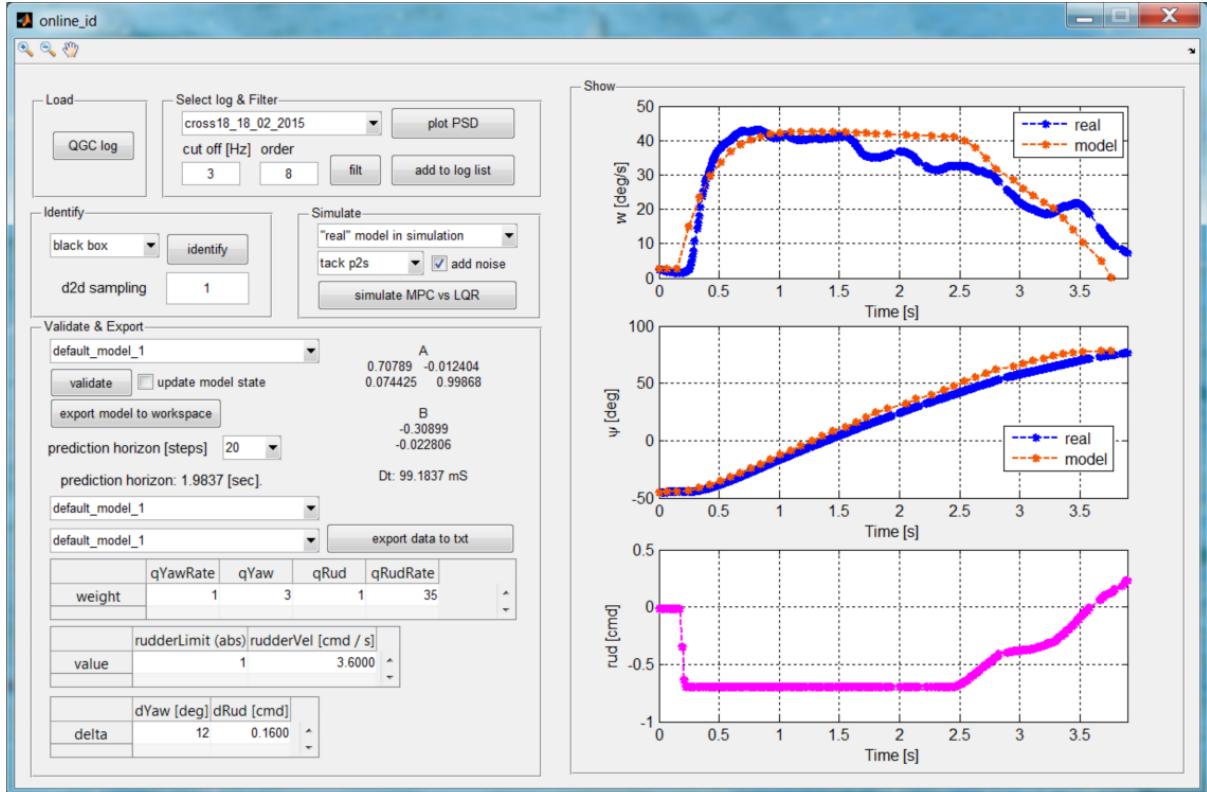


Figure 2.10: GUI developed for sailboat monitoring

2.4. Usages of USV

USV's has enabled data acquisitions that was thought to be unobtainable. Saildrone [7], founded in 2014, a company that specialize in the gathering of ocean data. The USV deployed by Saildrone is illustrated in Figure 2.11. The data that the USV gathers ranges from ocean temperature, humidity, pressure, wind speed and direction.

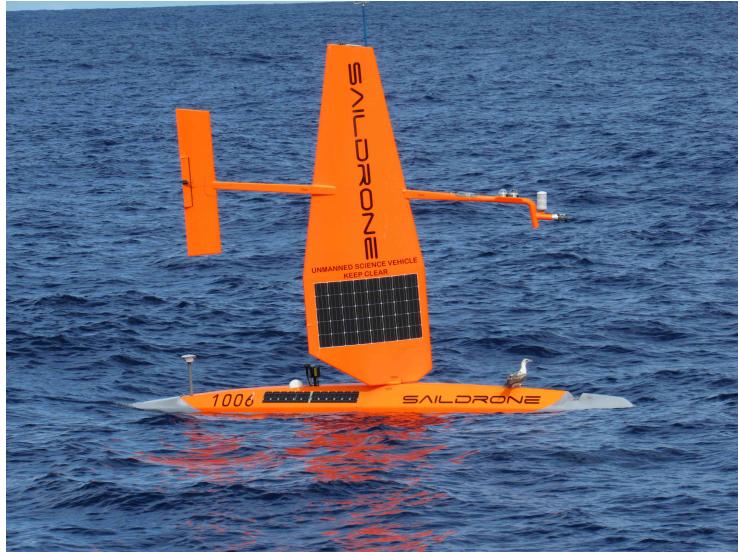


Figure 2.11: Saildrone's USV

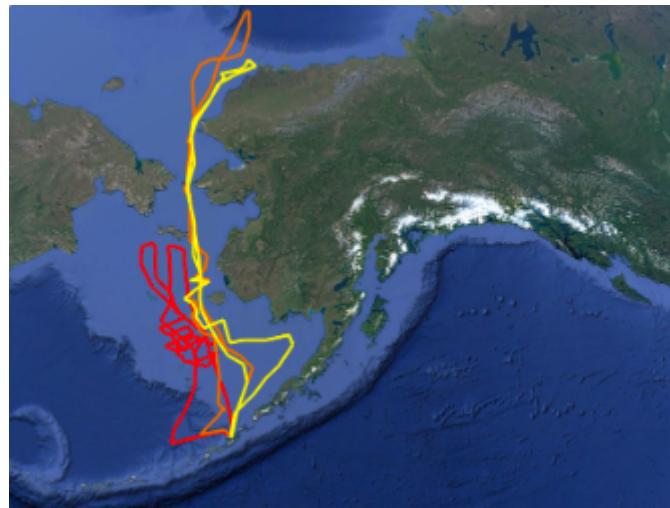


Figure 2.12: Routes of the USV's in Arctic Mission

USV's also allow for exploration of unknown ocean area's, with accompanied sensor USV's can also map out the ocean floor. One of the recent missions completed by Saildrone, was the 2017 Arctic Mission. Saildrone deployed three USV's from the Dutch Harbor in Alaska. Two of the vehicles were equipped with the CO₂ measuring devices and camera's for capturing video and photo footage of northern fur seals. The third vehicle is equipped with an echo sounder for surveying walleye pollock, nothern fur seals and the elusive North Pacific right whale. The three routes of the vehicles are illustrated in Figure 2.12. Saildrone is one of few companies/organisations that has developed the capable technology to make use of USV. To this extent a lot of research is still needed to develop USV and also is needed to utilise the data in the oceans. This is especially true in undeveloped countries and the ocean's surrounding these countries.

2.5. Ardupilot

Chapter 3

System Overview

This chapter gives an overview of the system used for developing the control strategies and details the platform used for testing the developed control strategies, the detailed hardware and software platform. The hardware controller used is a Pixhawk 6C flight controller, which is generally used for airplanes, however the software is Ardupilots rover firmware which has a basic implemented framework for a sailboat.

3.1. Physical System

The physical system consists of two parts: the sailboat and the ground station. The block diagram showcasing the system is illustrated in Figure 3.2. The sailboat used in testing the control is a dragonflite 95 model RC boat [8]. The full specifications of the sailboat is shown in Table A.4. Below, in Figure 3.1, is a photo of the dragonflite sailboat. The sailboat has a mainsail and a jib sail, the sails are controlled via a servo motor that controls the sail angle. The main and jib sail cannot be operated at different angles. For steering control another servo motor is controlling the rudder angle.



Figure 3.1: Dragonflite 95 RC model sailboat

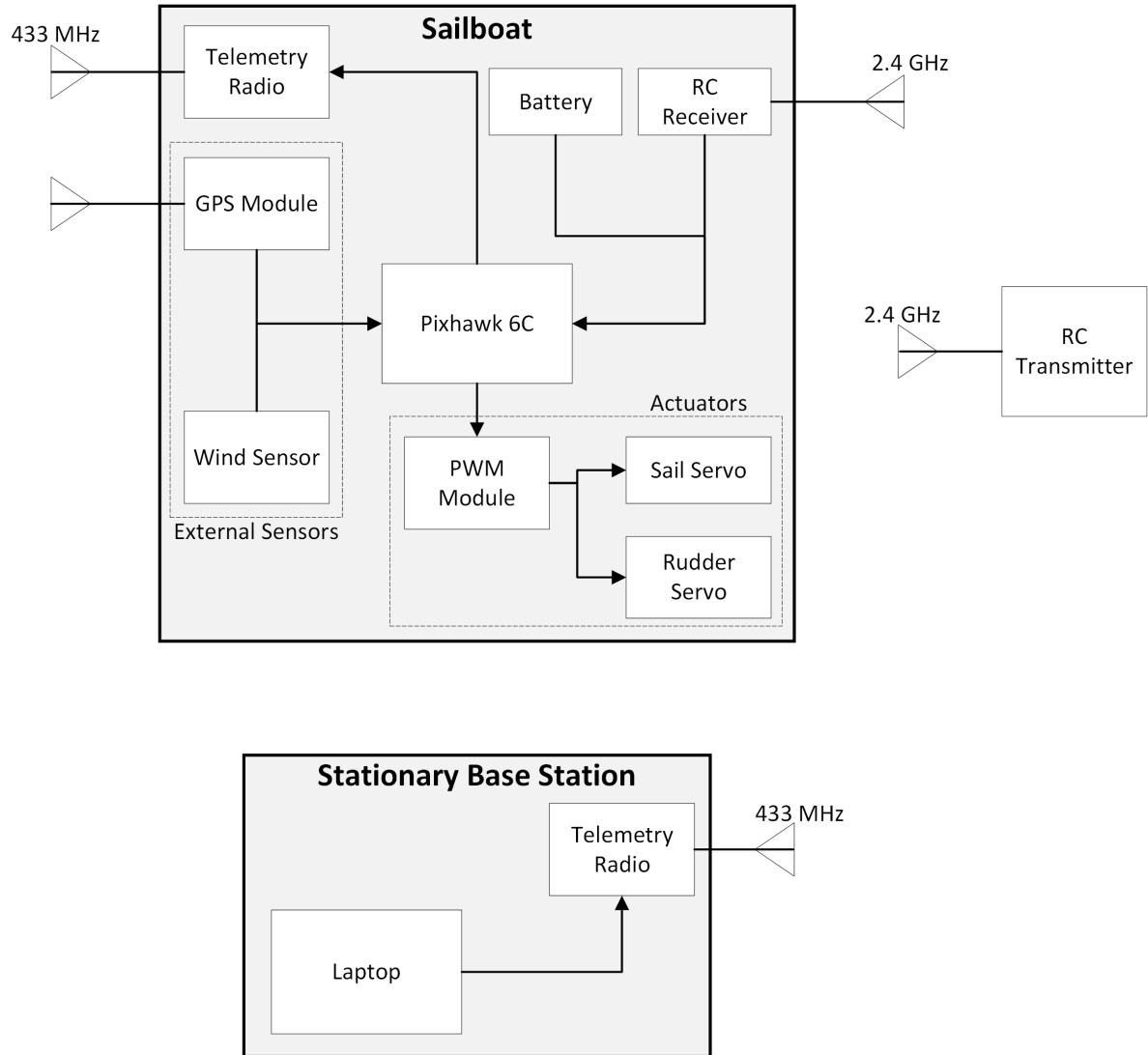


Figure 3.2: Block diagram overview of physical system

The block diagram illustrates a high-level overview of sensors, actuators and communications. The Sailboat receives manual control via the RC receiver and communicates with the base station via the telemetry radio. The Ground-Control-Software(GCS) running on the Laptop is Mission Planner [9] and the autopilot running on the Pixhawk is Ardupilot.

3.2. Marine Hardware

The marine hardware required for controlling and communicating with, the sailboat is illustrated in a block diagram in Figure 3.3. The hardware consists of off-the shelf hardware and a self designed wind sensor. The components are listed below,

- A Pixhawk 6C controller to run the autopilot software

- M8N GPS sensor supplying the controller with GPS data and compass data
- RC FS-iA6B receiver and FS-i6X transmitter to manually control sailboat
- Two servo motors, one for controlling sail servo and one for controlling the rudder
- PWM module that provides power to the servo motors and also transmit the PWM signals to the servos
- Two SiK telemetry radios for communicating between sailboat and base station
- PM02 V3 battery module used to monitor and regulate power supply to the Pixhawk
- Self designed wind sensor utilizing a rotary position hall effect sensor

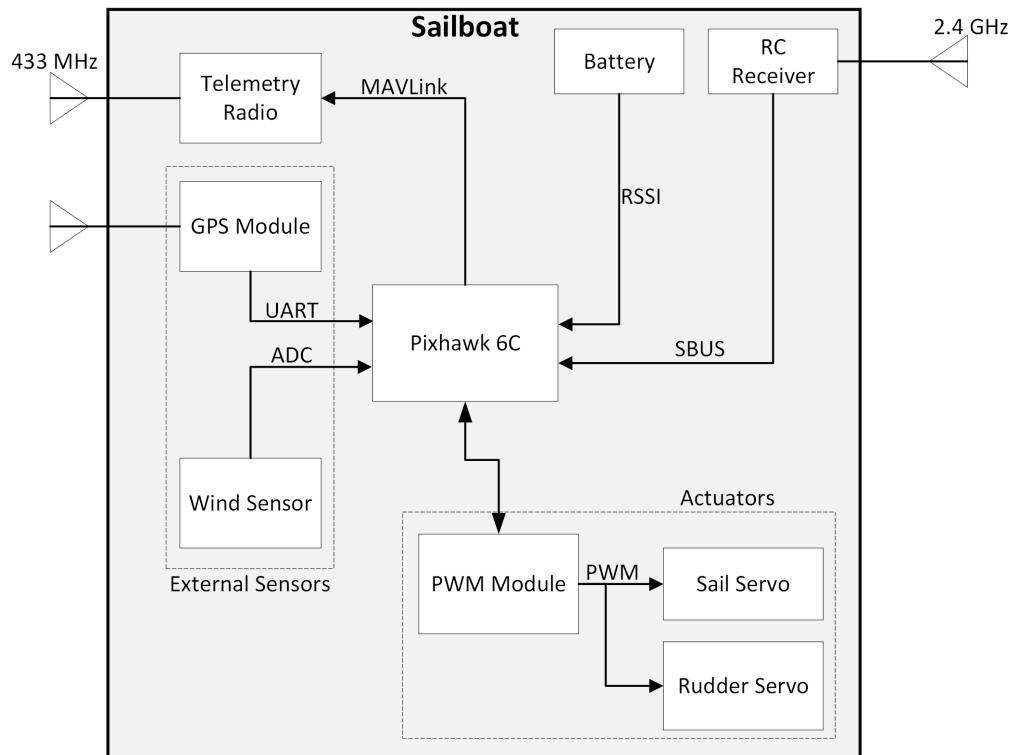


Figure 3.3: Block diagram of marine hardware

Each component is discussed in its own sub-section that shortly describes the component and references their data sheets.

3.2.1. Autopilot Controller

The hardware based around the Pixhawk 6C flight controller [10]. Traditionally the controller is used for airplanes and drones(copters). But there has been a recent shift to

use the controller for rovers and boats. The technical specifications of the Pixhawk 6C is shown in Chapter A.7, the dimensions of the Pixhawk 6C is illustrated in Figure A.1 and the pinout diagram of the Pixhawk 6C is illustrated in Figure A.2.



Figure 3.4: Pixhawk 6C

3.2.2. GPS and Compass Sensor

In addition to the flight controller the platform requires more sensors to allow for the autonomous control of the sailboat. The system requires a GPS and compass sensor, both these sensors are inside the M8N GPS sensor from Holybro [11]. The sensor is illustrated in Figure 3.5. The technical and feature specifications are shown in Chapter A.8 and the pinout diagram is illustrated in Figure A.3.



Figure 3.5: M8N GPS

3.2.3. Telemetry System

The telemetry system is required to communicate with the sailboat during missions and testing. The radios communicate via the MAVLink protocol [12], communication protocol and setup is explained in the Chapter ??.



Figure 3.6: Telemetry radio

The high level overview of the telemetry system is illustrated in Figure 3.7, one of the telemetry radios will plug into the PC, that runs the ground station software and communicate via serial protocol. The other telemetry radio will be onboard the sailboat. The two radios will communicate via the MAVLink protocol.



Figure 3.7: High Level Communication

3.2.4. Wind Vane

Due to the limited space on the sailboat and the importance of knowing what the wind direction is, a custom wind vane will be built to result in the best accuracy.



Figure 3.8: Caption

3.3. Stationary Base-Station

As mentioned the stationary base station consists of a laptop running a GCS called Mission Planner [9]. The laptop communicates with the sailboat via the MAVLink protocol. The block diagram of the base station is illustrated in Figure 3.9.

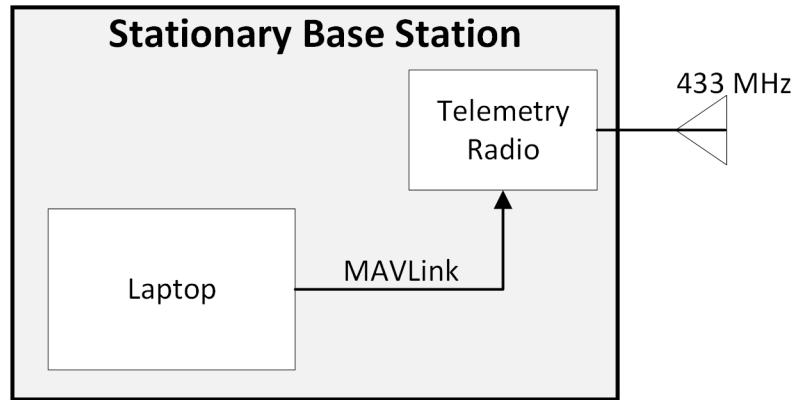


Figure 3.9: Block diagram of base station

3.3.1. Ground Control Software

Mission Planner is specifically designed to work in conjunction with the ArduPilot firmware. The main reason for using this GCS can be summarized below,

- Loading firmware onto flight controller boards

- Setup and configure parameters
- Tuning control systems of vehicle
- Setup missions, that consists of waypoints and fences, that you can load onto the autopilot
- Able to download and view logs from missions.
- SITL/HITL simulation capabilities

Figure 3.10 illustrates the Mission Planner GUI. On the right hand side is the maps, the upper left the Heads-Up Display(HUD) and bottom left is the toggle screen, which can be switched to display different screens. Ranging from the Actions screen, Messages and Dataflash Logs. At the top of the GUI there is tabs that lets you switch between different views, the one shown in Figure 3.10 is the **FLIGHT DATA** view, the **FLIGHT PLAN** view is where the mission waypoints and fences can be set.



Figure 3.10: Mission Planner GUI

The **SETUP** view is where you load the firmware onto your board, the **CONFIGURATION** view is where parameters in the autopilot can be changed and written to the flight controller and the last important view **SIMULATION** is where either the SITL or HITL simulation can be executed from.

Setup and Connecting

Mission Planning

The first part in executing a mission is to plan accordingly. Inside the **FLIGHT PLAN** view navigate the map until you find the area where the mission should be executed. Now at the top right of the map switch from mission to fence in the drop down box. Place the fence around the area you wish the vehicle to stay within. In the case of sailboat, this would be the area of water. The fence is created by placing points at the banks that connect in the form of a polygon. Now switch back to mission in the drop down box. The waypoints are placed the same way as the fence but instead of forming a polygon the waypoints will be numbered in the order that they are placed, take this into account and how it will effect the mission. The last thing to configure before writing the mission to the autopilot is to set the home point of the mission which will be where the sailboat is launched from. An example of a mission is illustrated in Figure 3.11.



Figure 3.11: An example of a mission consisting of waypoints and a fence

3.4. Simulation Software

3.4.1. Matlab Simulator

The control strategies were developed and tested on a non-linear simulation of a sailing yacht developed by Lin Xiao and Jerome Jouffroy [13]. The simulation was developed using Fossen's compact notation for marine vehicles and extending it to sailing yachts in the 4-DOF dynamic model. The simulator comes with a real time visualization of the sailboat and wind angle, illustrated in Figure 3.12.

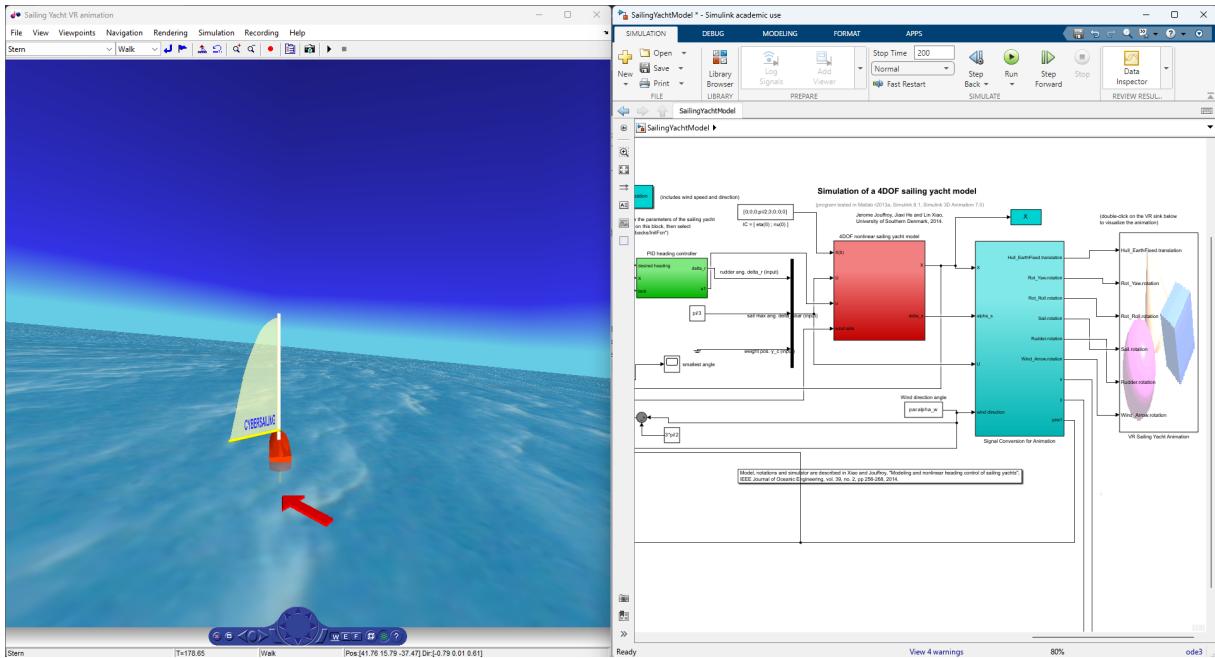


Figure 3.12: Overview of Matlab simulator

Once the sailboat platform is finished and practical test to determine its parameters are done the idea is to change the above mentioned simulator to more accurately present that of our sailboat. Then the simulator can also be used for future development. Also expanding the simulator to feature the effect of waves and ocean currents to more accurately represent real world scenarios.

3.4.2. ArduPilot Simulation

To test the Ardupilot software alongside with the modifications. Ardupilot have a SITL and HITL simulation for each of their frameworks, i.e. the sailboat. The simulation is a python script that mimics as if it is a Pixhawk so that the framework can be tested. The SITL and HITL allows for full functionality testing of the software on the autopilot. The architecture of the SITL is illustrated in Figure 3.13.

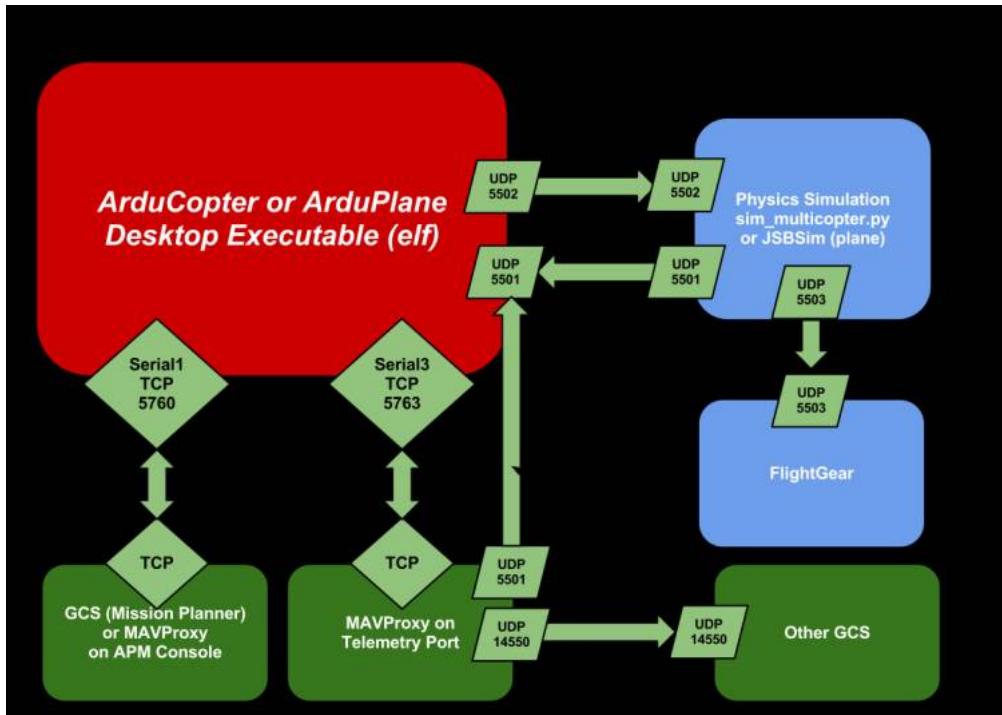


Figure 3.13: Architecture of Ardupilots SITL

Chapter 4

Ocean Vessel Dynamic Model

This chapter models a standard ocean vessel in six degrees of freedom, the hydrodynamic forces, the restoration forces and the external forces(wind,waves and ocean currents) acting on an ocean vessel, as defined in [14]. It then extends the modelling to a sailboat's keel, rudder and sail. The chapter ends of with a linearizations of the steering model and simplifications of the forces.

4.1. Reference Frames and Notations

It is convenient in the analysis of the motion for marine vessels in 6 DOF to define two Earth-centered coordinate frames, illustrated in Figure 4.1. In addition several other geographic reference frames are needed.

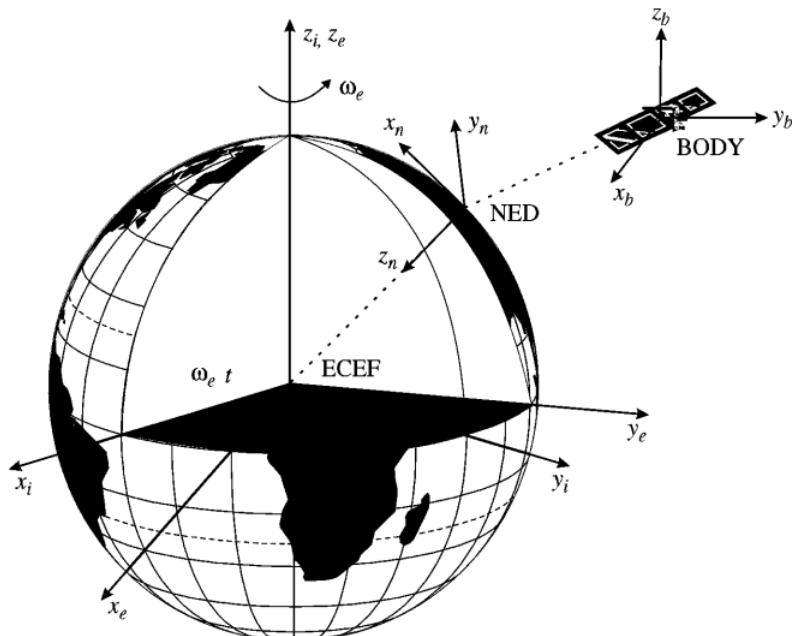


Figure 4.1: The Earth-centered Earth-fixed (ECEF) frame and an Earth-centered inertial (ECI) frame

4.1.1. Ocean Vessel Notation

An ocean vessel is modelled in six degrees of freedom, requiring six independent coordinates to determine its position and orientation. The first three coordinates corresponding to position (x, y, z) and their first time derivatives, translation motion along the x -, y -, and z -axes. The last three coordinates (ϕ, θ, ψ) and their first time derivatives describing orientation and rotational motion [15]. Figure 4.2 illustrates the motion variables of an ocean vessel with the six independent coordinates.

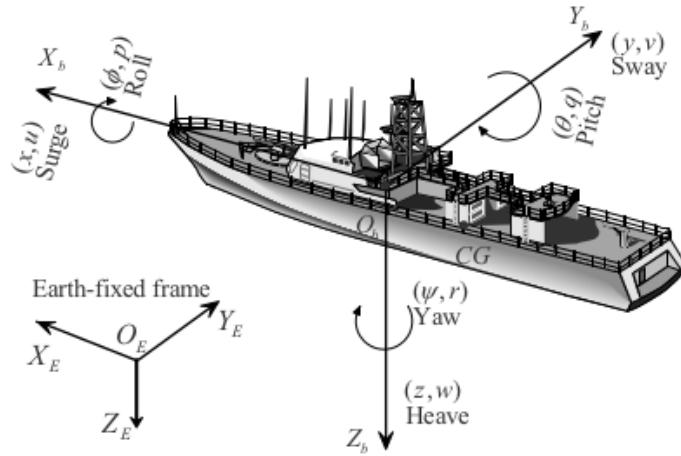


Figure 4.2: Motion variables for an ocean vessel

The Society of Naval Architects and Marine Engineers (SNAME) established the notation for the six different motion components as surge, sway, heave, roll, pitch and yaw. Table 4.1 summarizes the SNAME notation for ocean vessels.

Table 4.1: SNAME Notation for Ocean Vessels

DOF		Forces and Moments	Linear and angular velocity	Position and Euler angles
1	motion in the x direction (surge)	X	u	x
2	motion in the y direction (sway)	Y	v	y
3	motion in the z direction (heave)	Z	w	z
4	rotation in the x axis (roll)	K	p	ϕ
5	rotation in the y axis (pitch)	M	q	θ
6	rotation in the z axis (yaw)	N	r	ψ

4.1.2. Earth-Centered Reference Frames

Earth-centered Inertial Frame

The ECI is an inertial frame $\{i\}$ for terrestrial navigation, that is a non-accelerating reference frame in which Newton's laws of motion apply. This includes inertial navigation

system. The origin of $\{i\} = (x_i, y_i, z_i)$ is located at the centre o_i of the Earth with axes shown in Figure 4.1.

Earth-Centred Earth-Fixed Frame

The ECEF is a reference frame $\{e\} = (x_e, y_e, z_e)$ with its origin o_e fixed to the centre of the Earth but the axes rotate relative to the inertial frame ECI, which is fixed in space. For low speed marine vessels the earth's rotation is neglected and hence $\{e\}$ is considered to be inertial.

4.1.3. Geographic Reference Frame

North-East-Down Frame

The North-East-Down (NED) coordinate system $\{n\} = (x_n, y_n, z_n)$ with origin o_n is defined relative to the Earth's reference ellipsoid. It is defined as the tangent plane on the surface of the Earth moving with the vessel, but with axes pointing in different direction than the body-fixed axes of the vessel. The x axis points towards true *North*, the y axis points towards *East* while the z axis points downwards to the Earth's surface. The location of $\{n\}$ relative to $\{e\}$ is determined by using two angles l and μ denoting the *longitude* and *latitude*. For marine vessels operating in a local area, approximately constant longitude and latitude, an Earth-fixed tangent plane on the surface is used for navigation. This is referred to as flat Earth navigation and it will for simplicity be denoted by $\{n\}$. For flat Earth navigation one can assume that $\{n\}$ is inertial such that Newton's laws still apply.

Body Frame

The body-fixed reference frame $\{b\} = (x_b, y_b, z_b)$ with origin o_b is a moving coordinate frame that is fixed to the vessel, illustrated in Figure 4.2. The position and orientation of the vessel are described relative to the inertial reference frame while the linear and angular velocities of the vessel should be expressed in the body-fixed coordinate system. The origin o_b is usually chosen to coincide with a point midships in the water line. For ocean vessels the body axes are chosen to coincide with the *principle axes of inertia*, illustrated in Figure 4.2 and they are usually defined as

- x_b - longitudinal axis (directed from aft to fore)
- y_b - transversal axis (directed to starboard)
- z_b - normal axis (directed from top to bottom)

4.2. Rigid-Body Equations of Motion

4.2.1. Kinetics

The three translational motion kinetic equations and the three rotational kinetic equations are presented below, the full derivation is found in [16],

$$m[\dot{u} - vr + wq - x_g(q^2 + r^2) + y_g(pq - \dot{r}) + z_g(pr + \dot{q})] = X \quad (4.1)$$

$$m[\dot{v} - wp + ur - y_g(r^2 + p^2) + z_g(qr - \dot{p}) + x_g(qp + \dot{r})] = Y \quad (4.2)$$

$$m[\dot{w} - uq + vp - z_g(p^2 + q^2) + x_g(rp - \dot{q}) + y_g(rq + \dot{p})] = Z \quad (4.3)$$

$$\begin{aligned} I_x \dot{p} + (I_z - I_y)qr - (\dot{r} + pq)I_{xz} + (r^2 - q^2)I_{yz} + (qr - \dot{q})I_{xy} \\ + m[y_g(\dot{w} - qu + vp) - z_g(\dot{v} - wp + ut)] = K \end{aligned} \quad (4.4)$$

$$\begin{aligned} I_y \dot{q} + (I_x - I_z)rp - (\dot{p} + qr)I_{xy} + (p^2 - r^2)I_{zx} + (qp - \dot{r})I_{yz} \\ + m[z_g(\dot{u} - vr + wq) - x_g(\dot{w} - uq + vp)] = M \end{aligned} \quad (4.5)$$

$$\begin{aligned} I_z \dot{r} + (I_y - I_z)pq - (\dot{q} + rp)I_{yz} + (q^2 - p^2)I_{xy} + (rq - \dot{p})I_{zx} \\ + m[x_g(\dot{v} - wp + ur) - y_g(\dot{u} - vr + wq)] = N \end{aligned} \quad (4.6)$$

The equations stated above can be expressed in a vectorial setting as,

$$\mathbf{M}_{RB}\dot{\mathbf{v}} + \mathbf{C}_{RB}(\mathbf{v})\mathbf{v} = \boldsymbol{\tau}_{RB} \quad (4.7)$$

where $\mathbf{v} = [u \ v \ w \ p \ q \ r]^T$ is the generalized velocity vector decomposed in the body-fixed frame and $\boldsymbol{\tau}_{RB} = [X \ Y \ Z \ K \ M \ N]^T$ is the generalized vector of external forces and moments. The rigid body system inertia matrix \mathbf{M}_{RB} and the rigid body Coriolis and centripetal matrix \mathbf{C}_{RB} is defined in Equation A.2 and A.4. The generalized external force and moment vector, $\boldsymbol{\tau}_{RB}$, is equal to,

$$\boldsymbol{\tau}_{RB} = \boldsymbol{\tau}_H + \boldsymbol{\tau}_E + \boldsymbol{\tau} \quad (4.8)$$

where $\boldsymbol{\tau}_H$ is the hydrodynamic force and moment vector, $\boldsymbol{\tau}_E$ is the external disturbance force and moment vector and $\boldsymbol{\tau}$ is the propulsion force and moment vector.

4.2.2. Kinematics

Kinematics looks at the motion of the vessel without directly considering the forces affecting the motion. The first time derivative of the position vectors \mathbf{n}_1 and \mathbf{n}_2 is related to the linear velocity vector \mathbf{v}_1 and \mathbf{v}_2 via the following transformations,

$$\dot{\mathbf{n}}_1 = \mathbf{J}_1(\mathbf{n}_2)\mathbf{v}_1 \quad (4.9)$$

$$\dot{\mathbf{n}}_2 = \mathbf{J}_2(\mathbf{n}_2)\mathbf{v}_2 \quad (4.10)$$

where $\mathbf{J}_1(\mathbf{n}_2)$ and $\mathbf{J}_2(\mathbf{n}_2)$ are transformation matrices, which is related through the functions of the Euler angles: roll(ϕ), pitch(θ) and yaw(ψ). The \mathbf{J}_1 transformation matrix is given by

$$\mathbf{J}_1(\mathbf{n}_2) = \begin{bmatrix} \cos(\psi) \cos(\theta) & -\sin(\psi) \cos(\theta) + \sin(\phi) \sin(\theta) \cos(\psi) & \sin(\psi) \sin(\phi) + \sin(\theta) \cos(\psi) \cos(\phi) \\ \sin(\psi) \cos(\theta) & \cos(\psi) \cos(\phi) + \sin(\phi) \sin(\theta) \sin(\psi) & -\cos(\psi) \sin(\phi) + \sin(\theta) \sin(\psi) \cos(\phi) \\ -\sin(\theta) & \sin(\phi) \cos(\theta) & \cos(\phi) \cos(\theta) \end{bmatrix} \quad (4.11)$$

and the transformation matrix \mathbf{J}_2 is given by,

$$\mathbf{J}_2(\mathbf{n}_2) = \begin{bmatrix} 1 & -\sin(\phi) \tan(\theta) & \cos(\phi) \tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) / \cos(\theta) & \cos(\phi) / \cos(\theta) \end{bmatrix} \quad (4.12)$$

When $\theta = \pi/2$, the transformation matrix $\mathbf{J}_2(\mathbf{n}_2)$ becomes singular, however this is unlikely to happen when practically testing an ocean vessel, because of the metacentric restoring forces. Combining Equation 4.11 and Equation 4.12 results in the kinematics of an ocean vessel.

$$\begin{bmatrix} \dot{\mathbf{n}}_1 \\ \dot{\mathbf{n}}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{J}_1(\mathbf{n}_2) & 0_{3 \times 3} \\ 0_{3 \times 3} & \mathbf{J}_2(\mathbf{n}_2) \end{bmatrix} \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{bmatrix} = \dot{\mathbf{n}} = \mathbf{J}(\mathbf{n})\mathbf{v} \quad (4.13)$$

4.3. Hydrodynamic and Hydrostatic Forces and Moments

Hydrodynamic forces and moments can be defined as the forces and moments on a ocean body when the body is forced to oscillate with the wave excitation and no wave are incident on the body. As shown in [17], the hydrodynamic forces and moments acting on a rigid body can be assumed to be linearly superimposed. The forces and moments can be subdivided into three components,

1. Added mass due to the inertia of the surrounding fluid

2. Radiation-induced potential damping due to the energy carried away by the generated surface waves
3. Restoring forces due to Archimedian forces

The hydrodynamic forces and moments vector $\tau_{\mathbf{H}}$ is expressed in the equation below,

$$\tau_{\mathbf{H}} = -\mathbf{M}_{\mathbf{A}} \dot{\mathbf{v}} - \mathbf{C}_{\mathbf{A}}(\mathbf{v})\mathbf{v} - \mathbf{D}(\mathbf{v})\mathbf{v} - \mathbf{g}(\mathbf{n}) \quad (4.14)$$

where $\mathbf{M}_{\mathbf{A}}$ is the added mass matrix, $\mathbf{C}_{\mathbf{A}}(\mathbf{v})$ is the hydrodynamic Coriolis and centripetal matrix, $\mathbf{D}(\mathbf{v})$ is the damping matrix and $\mathbf{g}(\mathbf{n})$ is the position and orientation depending vector of restoring forces and moments. The added mass $\mathbf{M}_{\mathbf{A}}$ is given below,

$$\mathbf{M}_{\mathbf{A}} = \begin{bmatrix} X_{\dot{u}} & X_{\dot{v}} & X_{\dot{w}} & X_{\dot{p}} & X_{\dot{q}} & X_{\dot{r}} \\ Y_{\dot{u}} & Y_{\dot{v}} & Y_{\dot{w}} & Y_{\dot{p}} & Y_{\dot{q}} & Y_{\dot{r}} \\ Z_{\dot{u}} & Z_{\dot{v}} & Z_{\dot{w}} & Z_{\dot{p}} & Z_{\dot{q}} & Z_{\dot{r}} \\ K_{\dot{u}} & K_{\dot{v}} & K_{\dot{w}} & K_{\dot{p}} & K_{\dot{q}} & K_{\dot{r}} \\ M_{\dot{u}} & M_{\dot{v}} & M_{\dot{w}} & M_{\dot{p}} & M_{\dot{q}} & M_{\dot{r}} \\ N_{\dot{u}} & N_{\dot{v}} & N_{\dot{w}} & N_{\dot{p}} & N_{\dot{q}} & N_{\dot{r}} \end{bmatrix} \quad (4.15)$$

The hydrodynamic Coriolis and centripetal matrix is given below,

$$\mathbf{C}_{\mathbf{A}}(\mathbf{v}) = \begin{bmatrix} 0 & 0 & 0 & 0 & -a_3 & a_2 \\ 0 & 0 & 0 & a_3 & 0 & -a_1 \\ 0 & 0 & 0 & -a_2 & a_1 & 0 \\ 0 & -a_3 & a_2 & 0 & -b_3 & b_2 \\ a_3 & 0 & -a_1 & b_3 & 0 & -b_1 \\ -a_2 & a_1 & 0 & -b_2 & b_1 & 0 \end{bmatrix} \quad (4.16)$$

where a_1 , a_2 , a_3 , b_1 , b_2 and b_3 are defined in Equations A.5, A.6, A.7, A.8, A.9 and A.10. The general hydrodynamic damping experienced by ocean vessels is the potential damping, skin friction, wave drift damping and damping due to vortex shedding. The hydrodynamic damping can be expressed in a general form as below,

$$\mathbf{D}(\mathbf{v}) = \mathbf{D} + \mathbf{D}_{\mathbf{n}}(\mathbf{v}) \quad (4.17)$$

where the linear damping matrix \mathbf{D} is given below,

$$\mathbf{D} = - \begin{bmatrix} X_u & X_v & X_w & X_p & X_q & X_r \\ Y_u & Y_v & Y_w & Y_p & Y_q & Y_r \\ Z_u & Z_v & Z_w & Z_p & Z_q & Z_r \\ K_u & K_v & K_w & K_p & K_q & K_r \\ M_u & M_v & M_w & M_p & M_q & M_r \\ N_u & N_v & N_w & N_p & N_q & N_r \end{bmatrix} \quad (4.18)$$

4.3.1. Restoring Forces and Moments

Static stability considerations due to restoring forces are usually referred to as *metacentric stability* in the hydrostatic literature. A metacentric stable vessel will resist inclinations away from its steady-state or equilibrium points in heave, roll and pitch. For surface vehicles, the restoring force will depend on the vessel's metacentric height, the location of *CG* and *CB*, as well as the shape and size of the water plane. Let $A_{\omega\rho}$ denote the water plane area and

$$GM_T = \text{transverse metacentric height}(m) \quad (4.19)$$

$$GM_L = \text{longitudinal metacentric height}(m) \quad (4.20)$$

The metacentric height GM_i , where $i \in T, L$, is the distance between the metacentre M_i and the *CG*, as shown in Figure 4.3.

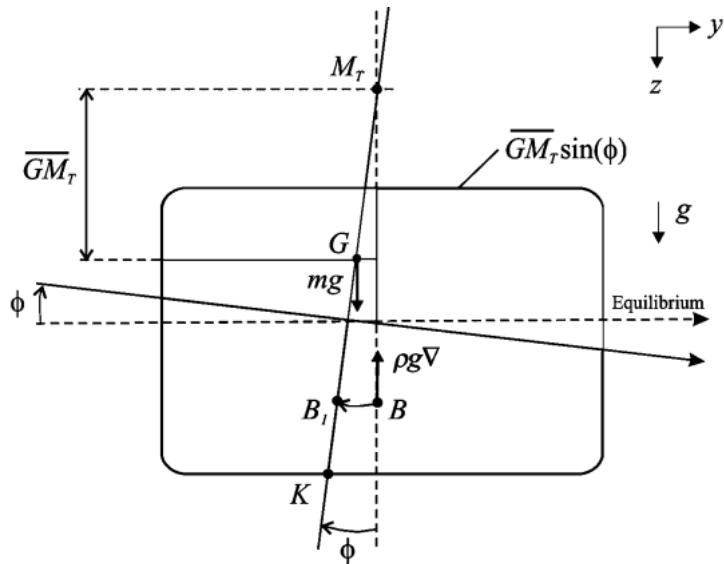


Figure 4.3: Transverse metacentric stability

For a floating vessel at rest, Archimedes stated that buoyancy and weight are in balance.

$$mg = pg\nabla \quad (4.21)$$

where m is the mass of the vessel, g gravitational force, ρ density of water and ∇ the nominal displaced water volume. Hence, the hydrostatic force in heave will be the difference between the gravitational and the buoyancy forces:

$$Z = mg - \rho g[\nabla + \delta\nabla(z)] = -\rho g\delta\nabla(z) \quad (4.22)$$

where the change in displaced water $\delta\nabla(z)$ is due to variations in heave position z . This can be written as

$$\delta\nabla(z) = \int_0^z A_{\omega\rho}(\delta) d\delta \quad (4.23)$$

where $A_{\omega\rho(\delta)}$ is the water plane area of the vessel as a function of the heave position. For conventional rigs and ships, however, it is common to assume that $A_{\omega\rho(\delta)} \approx A_{\omega\rho(0)}$ is constant for small perturbations in z . Hence, the restoring force Z will be linear in z , that is

$$Z \approx -\rho g A_{\omega\rho}(0)z \quad (4.24)$$

Recall that if a floating vessel is forced downwards by an external force such that $z \geq 0$, the buoyancy force becomes larger than the constant gravitational force since the submerged volume ∇ increases by $\delta\nabla$ to $\nabla + \delta\nabla$. This is physically equivalent to a spring with stiffness $Z_z = -\rho g A_{\omega\rho(0)}$ and position z . The restoring force expressed in body frame δf_r^b can therefore be written as

$$\delta f_r^b = -\rho g \begin{bmatrix} -\sin(\theta) \\ \cos(\theta)\sin(\phi) \\ \cos(\theta)\cos(\phi) \end{bmatrix} \int_0^z A_{\omega\rho}(\delta) d\delta \quad (4.25)$$

From Figure 4.3 it is seen that the moment arms in roll and pitch can be related to the moment arms $GM_T \sin(\phi)$ and $GM_L \sin(\theta)$ in roll and pitch and a z -direction force pair with magnitude $W = B = \rho g \nabla$. Therefore,

$$r_r^b = \begin{bmatrix} -GM_L \sin(\theta) \\ GM_T \sin(\phi) \\ 0 \end{bmatrix} \quad (4.26)$$

$$f_r^b = -0\rho g \nabla \begin{bmatrix} -\sin(\theta) \\ \cos(\theta)\sin(\phi) \\ \cos(\theta)\cos(\phi) \end{bmatrix} \quad (4.27)$$

By neglecting the moment contribution due to δf_r^b , consider only f_r^b , implies that the restoring moment becomes

$$m_r^b = r_r^b \times f_r^b = -\rho g \nabla \begin{bmatrix} GM_T \sin(\phi) \cos(\theta) \cos(\phi) \\ GM_L \sin(\theta) \cos(\theta) \cos(\phi) \\ (GM_T - GM_L \cos(\theta)) \sin(\phi) \sin(\theta) \end{bmatrix} \quad (4.28)$$

The assumption that $r_r^b \times \delta f_r^b = 0$ (no moments due to heave) is a good assumption since this term is small compared to $r_r^b \times f_r^b$. The restoring forces and moments are finally written as

$$\mathbf{g}(\mathbf{n}) = - \begin{bmatrix} \delta \mathbf{f}_r^b \\ \mathbf{m}_r^b \end{bmatrix} \quad (4.29)$$

$$\mathbf{g}(\mathbf{n}) = \begin{bmatrix} \rho g \int_0^z A_{\omega\rho}(\delta) d\delta \sin(\theta) \\ \rho g \int_0^z A_{\omega\rho}(\delta) d\delta \cos(\theta) \sin(\phi) \\ \rho g \int_0^z A_{\omega\rho}(\delta) d\delta \cos(\theta) \cos(\phi) \\ \rho g \nabla GM_T \sin(\phi) \cos(\theta) \cos(\phi) \\ \rho g \nabla GM_L \sin(\theta) \cos(\theta) \cos(\phi) \\ \rho g \nabla (GM_T + -GM_L \cos(\theta)) \sin(\phi) \sin(\theta) \end{bmatrix} \quad (4.30)$$

4.4. Environmental Disturbances

The forces and moments induced by the environmental disturbances is defined by the vector τ_E and includes ocean currents, waves(wind generated) and wind.

$$\tau_E = \tau_E^{cu} + \tau_E^{wa} + \tau_E^{wi} \quad (4.31)$$

where τ_E^{cu} , τ_E^{wa} and τ_E^{wi} are vectors of forces and moments induced by ocean currents, waves and wind.

4.4.1. Current-induced Forces and Moments

The current induced forces and moments vector τ_E^{cu} is given by

$$\tau_E^{cu} = (\mathbf{M}_{RB} + \mathbf{M}_A) \dot{\mathbf{v}}_c + \mathbf{C}(\mathbf{v}_r) \mathbf{v}_r - \mathbf{C}(\mathbf{v}) \mathbf{v} + \mathbf{D}(\mathbf{v}_r) \mathbf{v}_r - \mathbf{D}(\mathbf{v}) \mathbf{v} \quad (4.32)$$

where $\mathbf{v}_r = \mathbf{v} - \mathbf{v}_c$ and $\mathbf{v}_c = [u_c, v_c, w_c, 0, 0, 0]^T$ is a vector irrotational body-fixed current velocities. Take the earth-fixed velocity vector denoted by $[u_c^E, v_c^E, w_c^E]^T$, then the body fixed components $[u_c, v_c, w_c]^T$ can be calculated by

$$\begin{bmatrix} u_c \\ v_c \\ w_c \end{bmatrix} = \mathbf{J}_1^T(\mathbf{n}_2) \begin{bmatrix} u_C^E \\ v_C^E \\ w_C^E \end{bmatrix} \quad (4.33)$$

4.4.2. Wave-induced Forces and Moments

The vector τ_E^{wa} of the wave-induced forces and moments is given by

$$\tau_E^{wa} = \begin{bmatrix} \sum_{i=1}^N \rho g BL T \cos(\beta) s_i(t) \\ \sum_{i=1}^N \rho g BL T \sin(\beta) s_i(t) \\ 0 \\ 0 \\ 0 \\ \sum_{i=1}^N \frac{1}{24} \rho g BL (L^2 - B^2) \sin(2\beta) s_i^2(t) \end{bmatrix} \quad (4.34)$$

where β is the vessel's heading(encounter) angle, illustrated in Figure , ρ is the water density, L is the length of the vessel, B is the breadth of the vessel and T is the draft of the vessel. Ignoring the higher-order terms of the wave amplitude, the wave slope $s_i(t)$ for the wave component i is defined by

$$s_i(t) = A_i \frac{2\pi}{\lambda_i} \sin(\omega_{ei} t + \phi_i) \quad (4.35)$$

where A_i is the wave amplitude, λ_i is the wave length, ω_{ei} is the encounter frequency and ϕ_i is a random phase uniformly distributed and constant with time $[0, 2\pi)$ corresponding to the wave component i .

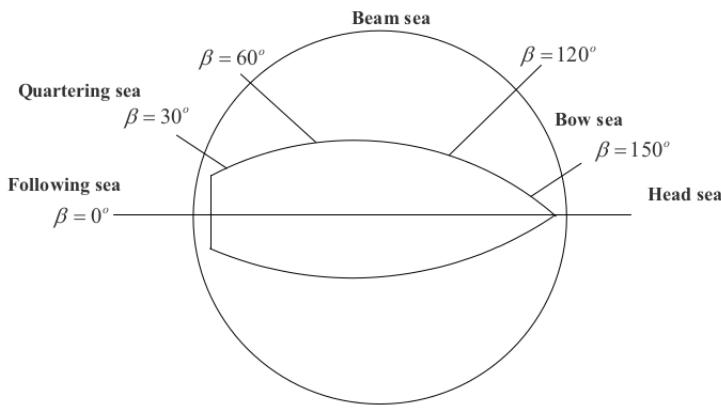


Figure 4.4: Ocean vessel's heading angle

4.4.3. Wind-induced Forces and Moments

When the ocean vessel is at rest the vector τ_E^{wi} of the wind induced forces and moments is given by

$$\tau_E^{\omega i} = \begin{bmatrix} C_X(\gamma_\omega)A_{F\omega} \\ C_Y(\gamma_\omega)A_{L\omega} \\ C_Z(\gamma_\omega)A_{F\omega} \\ C_K(\gamma_\omega)A_{L\omega}H_{L\omega} \\ C_M(\gamma_\omega)A_{F\omega}H_{F\omega} \\ C_N(\gamma_\omega)A_{L\omega}L_{oa} \end{bmatrix} \quad (4.36)$$

where V_ω is the wind speed, ρ_a is the air density, $A_{F\omega}$ is the frontal projected area, $A_{L\omega}$ is the lateral projected area, $H_{F\omega}$ is the centroid of $A_{F\omega}$ above the water line, $H_{L\omega}$ is the centroid of $A_{L\omega}$ above the water line, L_{oa} is the over all length of the vessel, γ_ω is the angle of relative wind of the vessel bow, illustrated in Figure 4.5 and is given by

$$\gamma_\omega = \psi - \beta_\omega - \pi \quad (4.37)$$

where β_ω being the wind direction. All the wind coefficients(look-up tables) $C_X(\gamma_\omega)A_{F\omega}$, $C_Y(\gamma_\omega)A_{L\omega}$, $C_Z(\gamma_\omega)A_{F\omega}$, $C_K(\gamma_\omega)A_{L\omega}H_{L\omega}$, $C_M(\gamma_\omega)A_{F\omega}H_{F\omega}$ and $C_N(\gamma_\omega)A_{L\omega}L_{oa}$ are computed numerically or by experiments in a wind tunnel as shown in [18].

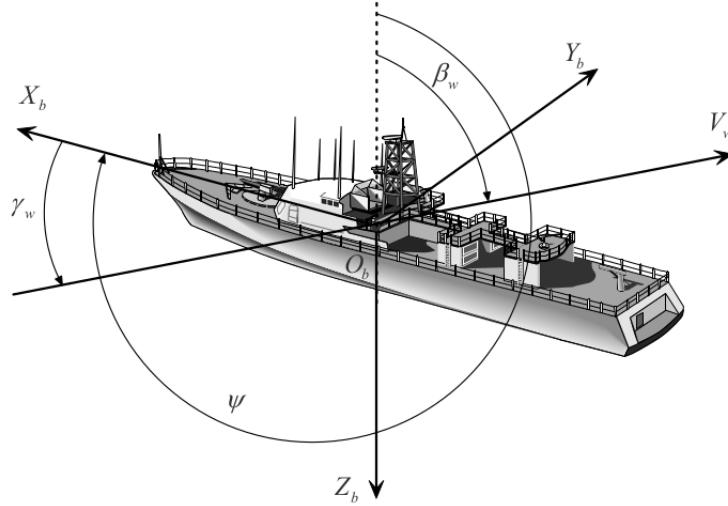


Figure 4.5: Wind angle on vessel

When the vessel is moving the vector $\tau_E^{\omega i}$ is given by

$$\tau_E^{\omega i} = \begin{bmatrix} C_X(\gamma_{r\omega})A_{F\omega} \\ C_Y(\gamma_{r\omega})A_{L\omega} \\ C_Z(\gamma_{r\omega})A_{F\omega} \\ C_K(\gamma_{r\omega})A_{L\omega}H_{L\omega} \\ C_M(\gamma_{r\omega})A_{F\omega}H_{F\omega} \\ C_N(\gamma_{r\omega})A_{L\omega}L_{oa} \end{bmatrix} \quad (4.38)$$

where

$$V_{r\omega} = \sqrt{u_{r\omega}^2 + v_{r\omega}^2} \quad (4.39)$$

$$\gamma_{r\omega} = -\arctan2(v_{r\omega}, u_{r\omega}) \quad (4.40)$$

with

$$u_{r\omega} = u - V_\omega \cos(\beta_\omega - \psi) \quad (4.41)$$

$$v_{r\omega} = v - V_\omega \cos(\beta_\omega - \psi) \quad (4.42)$$

4.5. Simplified Maneuvering Model including Roll(4 DOF)

The 6 DOF rigid-body kinetic equations can be further simplified to a 4 DOF model that describes the maneuvering model of a vessel and includes the roll, formulated in [19]. This model is commonly used in the simulation of nonlinear vessels and in the design of control systems. The 4 DOF equations of motions are,

$$\mathbf{M}\dot{\mathbf{v}} + \mathbf{C}_{RB}(\mathbf{v})\mathbf{v} + \mathbf{C}_A(\mathbf{v})\mathbf{v} + \mathbf{D}(\mathbf{v})\mathbf{v} + \mathbf{G}\mathbf{n} = \boldsymbol{\tau} + \boldsymbol{\tau}_{wind} + \boldsymbol{\tau}_{wave} + \boldsymbol{\tau}_{ocean} \quad (4.43)$$

where

$$\mathbf{M} = \begin{bmatrix} m - X_{\dot{u}} & 0 & 0 & 0 \\ 0 & m - Y_{\dot{v}} & -mz_g - Y_{\dot{p}} & mx_g - Y_{\dot{r}} \\ 0 & -mz_g - K_{\dot{v}} & I_x - K_{\dot{p}} & 0 \\ 0 & mx_g - N_{\dot{v}} & 0 & I_z - N_{\dot{r}} \end{bmatrix} \quad (4.44)$$

$$\mathbf{C}_{RB}(\mathbf{v}) = \begin{bmatrix} 0 & 0 & mz_g r & -m(x_g r + v) \\ 0 & 0 & 0 & mu \\ -mz_g & 0 & 0 & 0 \\ m(x_g r + v) & -mu & 0 & 0 \end{bmatrix} \quad (4.45)$$

$$\mathbf{C}_A(\mathbf{v}) = \begin{bmatrix} 0 & 0 & 0 & Y_{\dot{v}} v \\ 0 & 0 & 0 & -X_{\dot{u}} u \\ 0 & 0 & 0 & Y_{\dot{v}} v \\ -Y_{\dot{v}} v & X_{\dot{u}} u & -Y_{\dot{v}} v & 0 \end{bmatrix} \quad (4.46)$$

$$\mathbf{G} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -K_\phi & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (4.47)$$

4.6. Modelling of a Sail, Keel and Rudder

4.6.1. Rudder Theory

Rudder Theory and Terminology

The rudder of a ocean vessel is located at the rear of the vessel and is used for steering. The rudder is usually completely submerged and therefor the forces applicable to the rudder is only hydrodynamic. A rudder changes steers a ocean vessel by generating a lifting force that causes the vessel change its heading [20]. The rudder is fundamental to the safe operation of the ocean vessel. In the case of a sailboat a rudder is very important in establishing an angle of attack and adjusting the sail direction accordingly. The terminology associated with a lifting foil is shown below, where a rudder only differs with having no maximum camber, because it is symmetrical.

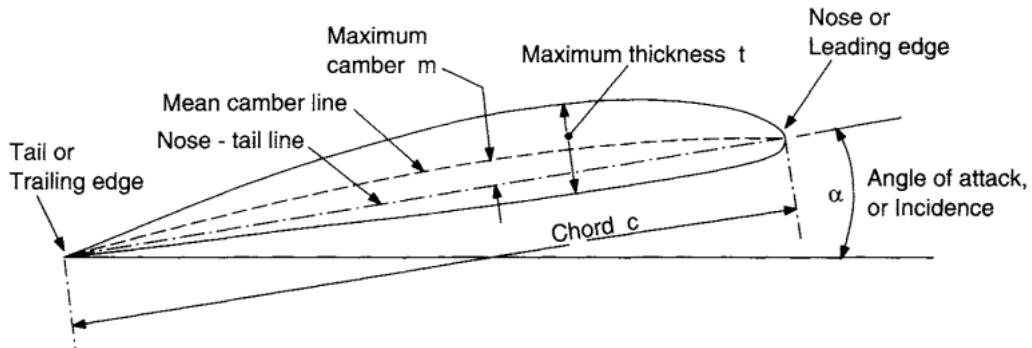


Figure 4.6: Terminology of a lifting foil

Hydrodynamics of a Rudder

The lifting action of the rudder arises from the difference in the average pressure of the fluid over the upper and lower surfaces of the rudder, illustrated in Figure 4.7.

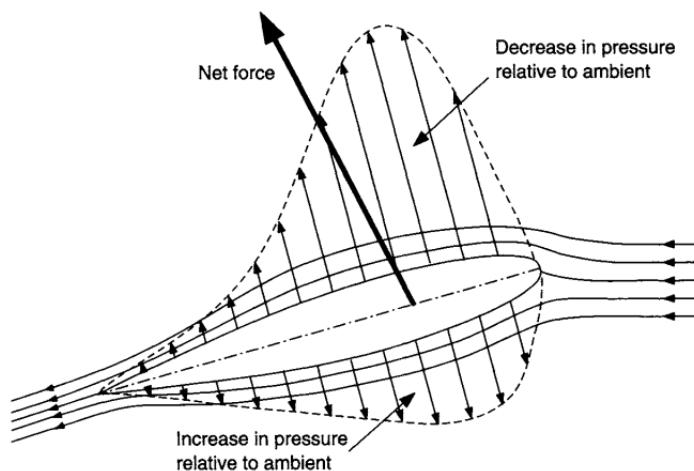


Figure 4.7: Pressure around the rudder

The total force on the rudder consists of a *lift* component perpendicular to the fluid stream U_0 and a *drag* component parallel to U_0 , illustrated in Figure 4.8.

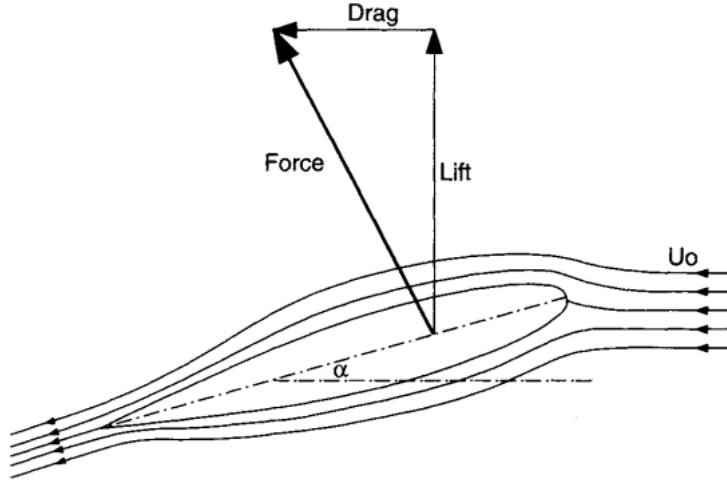


Figure 4.8: Lift and drag force on rudder

These components are usually presented in terms of dimensionless coefficients for a given rudder angle of attack α , where α is the angle between the rudder and the direction of flow.

$$C_L = \frac{Lift}{0.5\rho AU_0^2} \quad (4.48)$$

$$C_D = \frac{Drag}{0.5\rho AU_0^2} \quad (4.49)$$

where A is the rudder area, U_0 the fluid free-stream speed and ρ the fluid density. C_L and C_D depend on the rudder geometry, the incidence to the incoming fluid flow and the Reynolds number(Re).

Rudder Forces and Moments

The forces and moments experienced by a sailboat due to the rudder are defined in [21]. The equations formulated for the forces and moments are illustrated below

$$X_{rud} = C_{X\delta_R} \sin(\alpha_R) \sin(\delta_R) \times \frac{1}{2} \rho_w v_B^2 L_W D_K \quad (4.50)$$

$$Y_{rud} = C_{Y\delta_R} \sin(\alpha_R) \cos(\delta_R) \cos(\phi) \times \frac{1}{2} \rho_w v_B^2 L_W D_K \quad (4.51)$$

$$K_{rud} = C_{K\delta_R} \sin(\alpha_R) \sin(\delta_R) \times \frac{1}{2} \rho_w v_B^2 L_W D_K \quad (4.52)$$

$$N_{rud} = C_{N\delta_R} \sin(\alpha_R) \cos(\delta_R) \cos(\phi) \times \frac{1}{2} \rho_w v_B^2 L_{WL} D_K \quad (4.53)$$

where $C_{X\delta_R}$, $C_{Y\delta_R}$, $C_{K\delta_R}$, $C_{N\delta_R}$ are non-dimensional coefficients, V_B is the boat velocity, ρ_w is the water density, L_{WL} is the length on design waterline, D_K is the design draft length, δ_R is the physical rudder angle and α_R is the effective angle of attack on the rudder as defined below

$$\alpha_R = \delta_R - \epsilon_y \gamma - \tan^{-1} \left(\frac{x_R R}{U} \right) \quad (4.54)$$

$$\epsilon = \frac{d\epsilon}{d\gamma} \times \gamma = \epsilon_\gamma \gamma \quad (4.55)$$

where γ is the leeway angle the sailboat is sailing and ϵ is the angle of inflow from the downwash generated by the keel and x_R is the longitudinal distance of the quarter-chord point of the rudder to the CG of the boat. The angles α_R , δ_R , γ and ϵ are illustrated in Figure 4.9

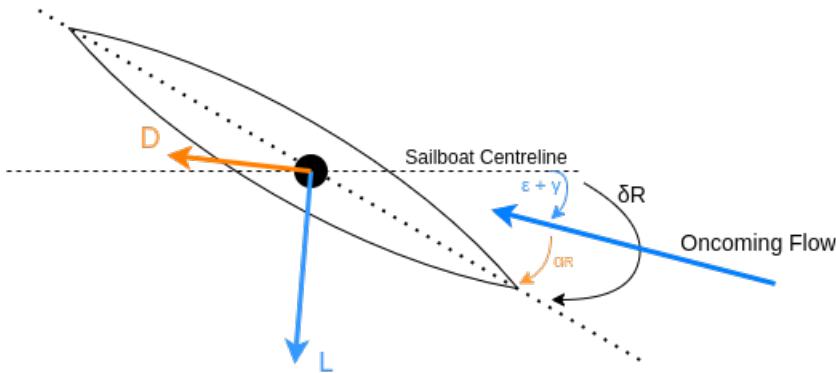


Figure 4.9: Definition of Rudder Angles

4.6.2. Sail Theory

Sail Theory and Terminology

Sail makers make use of their own language in naming for sails and sailboats [22]. It is important to know this language in order to have a conversation about sails. For instance the front of a sail is called the *bow* while the rear is called the *stern*. As shown in Figure 4.10 the *luff* is the leading edge of the sail and the *leech* is the trailing. The *foot* is the bottom edge which can be attached to the *boom* or left loose. Sails generally come to a point at the head, the attachment point to the mast at the top of the sail, but it is also common to see sails with a square top. The *tack* and the *clew* are the attachment points of the foot at the *luff* and *leech* respectively.

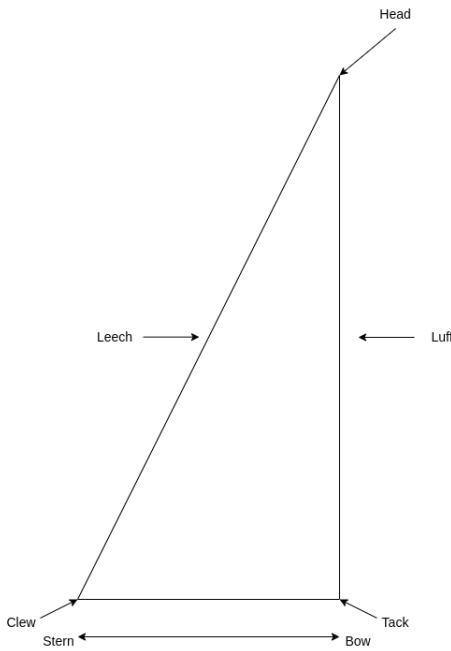


Figure 4.10: Naming convention of a sail

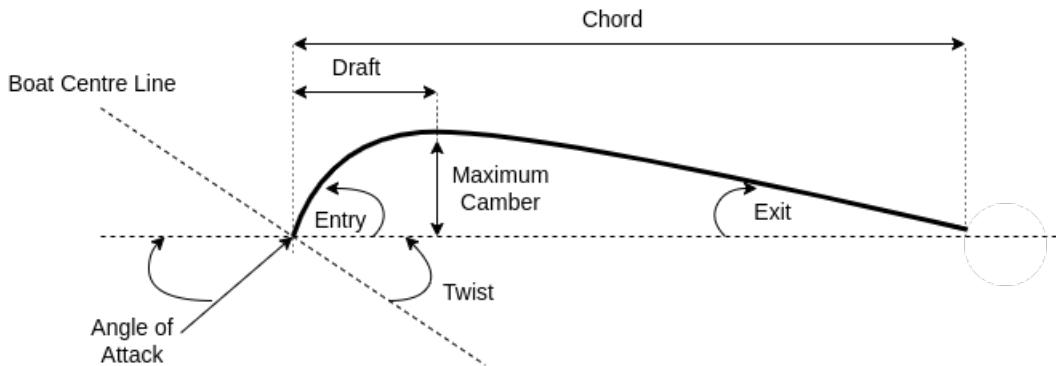


Figure 4.11: Terminology of a sail wing and shape

The shape of a sail can vary in its height and span and has almost no thickness, a horizontal section can be classified using the terminology of a wing. The terms are illustrated in Figure 4.11. The *chord* is the straight line between the leading and trailing edge. *Camber* is then the perpendicular distance from the chord line to the foil. *Draft* is the position of the maximum camber along the chord line. *Entry* and *exit* angles of the foil are, respectively, the angles of the leading and trailing edge to the chord line. Angle of attack(*AoA*) is the angle between the oncoming flow and the chord line and the *twist* angle is the angle between the chord line and the sailboat's centre line.

Depending on the tack of the boat, the side from which the wind is blowing from, illustrated in Figure 4.12, is the *windward* side and the other side will be *leeward* side.

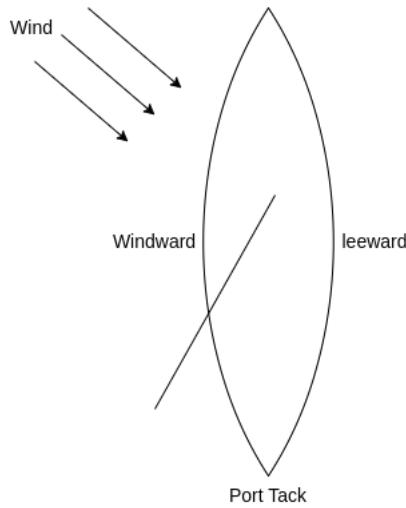


Figure 4.12: Windward and leeward definitions

Aerodynamics of a Sail

The way sail function, is much alike to that of a wing with some differences. Both a sail and a wing generate a force due to a pressure difference acting over their area. The three major differences between a sail and a wing are that a sail has almost no thickness, often has a large camber and generally operate at higher angles of attack. The flow around the sail is also usually disturbed by the mast or stays leading to large areas of separation as shown in [23]. Bernoulli's equation state that the pressure is inversely related to velocity and thus it can be seen through the contraction of the streamlines over the leeward side that the flow will accelerate creating a drop in pressure.

The simplified explanation of how lift is created due to the particles travelling further over the leeward side does not hold much substances for thin aerofoils such as sails. The way a airplane wing generates lift is due to the shape of the airfoil, the air flows faster over the top than it does over the bottom, because it has a further travel distance. With thin-airfoil sails the distance over the top and bottom is the same, so this the same reasoning for lift does not hold. By looking at the numerical solution of a flat plate, found in [24], gives insight to how a sail is affected by pressure. The first results, illustrated in Figure 4.13, showed that the thin-airfoil has no lift, which is incorrect.

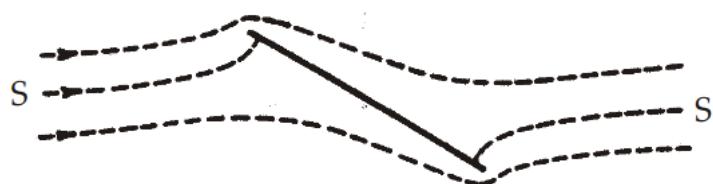


Figure 4.13: Flow field without circulation

Noting that these mathematical models determined streamlines that make very sharp

turns in getting around the leading edge and trailing edge of the airfoil. For a thin airfoil this would mean infinite velocities, and to reduce these velocities the airfoil can be bent around the leading edge. For the flow at the trailing edge the assumption is made that the airflow will leave the airfoil smoothly in a direction determined by an imaginary slight extension of the airfoil, which is known as the Kutta condition. It has been found that the Kutta condition can be satisfied mathematically by superimposing another type of flow solution, called circulation, onto the flat plate airfoil model. The superposition is shown in Figure 4.14.

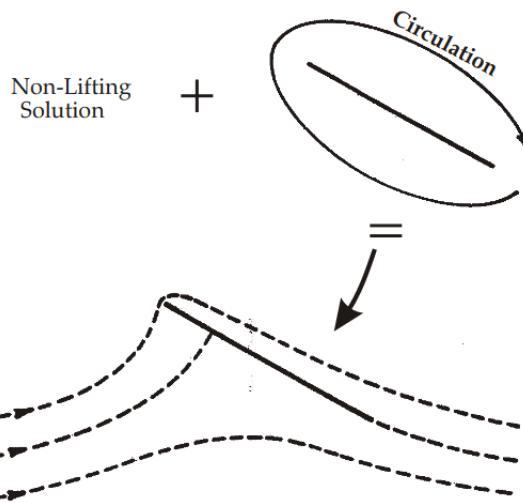


Figure 4.14: Superposition of circulation and non-circulation solution to give lift

Figure 4.15 illustrates the lift and drag component experienced due to wind on the sail. The drag component is always in the direction of the apparent wind and the lift is perpendicular to the apparent wind. γ_ω is the apparent wind angle, α is the AoA of the sail wing, ψ is the leeway angle, which is the angle between the course and the heading, and θ is the sinusoidal angle between the net and lift force.

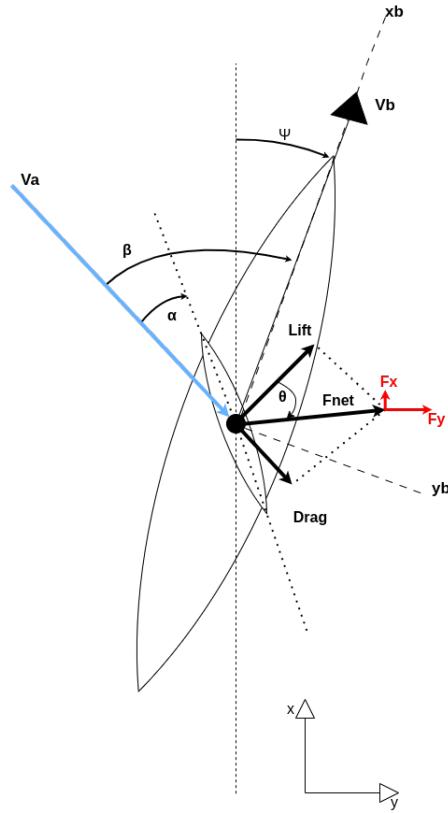


Figure 4.15: Lift and drag forces due to wind on sail wing

The lift and drag forces are defined by,

$$F_L = \frac{1}{2} C_L \rho U_A^2 A \quad (4.56)$$

$$F_D = \frac{1}{2} C_D \rho U_A^2 A \quad (4.57)$$

where C_L and C_D are the lift and drag coefficients, U_A the apparent wind speed, A area of the sail and ρ the air density. The lift and drag forces are combined into a single net aerodynamic force whose magnitude and angle relative to the apparent wind perpendicular defined as

$$|F_{net}| = F_{net} = \frac{1}{2} \rho U_A^2 A \sqrt{C_L^2 + C_D^2} \quad (4.58)$$

$$\theta = \tan^{-1} \left(\frac{C_D}{C_L} \right) \quad (4.59)$$

The component parallel to the direction of motion is the aerodynamic driving force F_x , while the force perpendicular to the direction of motion is the aerodynamic side force. Besides being a function of the angle of attack of the wing sail (θ is a function of α), F_x and F_y are both dependent on the apparent wind angle and the wind profile in which the vessel is sailing. F_x and F_y are defined below,

$$F_x = F_{net} \sin(\gamma_\omega - \theta) \quad (4.60)$$

$$F_y = F_{net} \cos(\gamma_\omega - \theta) \quad (4.61)$$

The equations above is valid for any lifting surface, whether it is a rigid wing sail or a cloth sail. For efficient performance of a sailing vehicle, the lifting surface will maximize the aerodynamic driving force while minimizing the aerodynamic side force. At a given leeway angle, the aerodynamic force ratio (F_x/F_y) is at a maximum when θ is minimized. In the limit that $C_D/C_L \rightarrow 0$, the aerodynamic force ratio approaches the limit

$$F_x/F_y \rightarrow \tan(\gamma_\omega)$$

Sail Forces and Moments

The sail forces and moments are expressed in their horizontal components. When the sail is in the upright position the aerodynamic coefficients X'_{s0} and Y'_{s0} are expressed using lift coefficient L'_{s0} and drag coefficient D'_{s0} as shown in [25].

$$X'_{s0} = L'_{s0} \sin(\gamma_\omega) - D'_{s0} \cos(\gamma_\omega) \quad (4.62)$$

$$Y'_{s0} = L'_{s0} \cos(\gamma_\omega) + D'_{s0} \sin(\gamma_\omega) \quad (4.63)$$

where the subscript of 0 means value at the upright condition.

In the heeled condition, when the vessel is rotated due to the sail lift and drag force, the effect of the heel on the aerodynamic forces is produced by the reduction of both the apparent wind angle and apparent wind speed. The apparent wind angle in the heeled condition $\gamma_{\omega\phi}$ is expressed as follows using the apparent wind angle γ_ω and apparent wind speed U_A :

$$\gamma_{\omega\phi} = \tan^{-1} \left(\frac{U_A \sin(\gamma_\omega) \cos(\phi)}{U_A \cos(\gamma_\omega)} \right) = \tan^{-1} (\tan(\gamma_\omega) \cos(\phi)) \quad (4.64)$$

The apparent wind speed in the heeled condition $U_{A\phi}$ is also expressed as:

$$U_{A\phi} = \sqrt{((U_A \cos(\gamma_\omega))^2 + (U_A \sin(\gamma_\omega) \cos(\phi))^2)} = U_A \sqrt{1 - (\sin(\gamma_\omega) \sin(\phi))^2} \quad (4.65)$$

For the close-hauled condition, tacking behaviour of sailboat due to not being able to sail directly into the wind, the sail may not stall due to the small attack angle. Therefore, the lift force will decrease proportionally to the reduction of both the apparent wind angle and the dynamic pressure of flow(square of the apparent wind speed). Hence the decreasing ratio of lift force by the heel angle ϕ can be described as:

$$\left(\frac{\gamma_{\omega\phi}}{\gamma_{\omega}}\right) \left(\frac{U_{A\phi}}{U_A}\right) = \left(\frac{\tan^{-1}(\tan(\gamma_{\omega})\cos(\phi))}{\gamma_{\omega}}\right) \left(1 - (\sin(\gamma_{\omega})\sin(\phi))^2\right) \quad (4.66)$$

The vector of lift force inclines with heel angle and rotates in the normal plane to the apparent wind axis. Since the angle between the apparent wind axis and the boat center line(heeling axis) is γ_{ω} , the rotating angle of the lift force vector ϕ' in the normal plane to the apparent wind axis is given by:

$$\phi' = \sin^{-1}(\cos(\gamma_{\omega})\sin(\phi)) \quad (4.67)$$

Therefore, the decreasing ratio of horizontal component of the lift force is expressed as:

$$\left(\frac{\gamma_{\omega\phi}}{\gamma_{\omega}}\right) \left(\frac{U_{A\phi}}{U_A}\right) \cos(\phi') = \left(\frac{\tan^{-1}(\tan(\gamma_{\omega})\cos(\phi))}{\gamma_{\omega}}\right) \left(1 - (\sin(\gamma_{\omega})\sin(\phi))^2\right) \cos(\sin^{-1}(\cos(\gamma_{\omega})\sin(\phi))) \quad (4.68)$$

Expanding the above mention equation in a power series and assuming that γ_A is small, results in

$$\left(\frac{\gamma_{\omega\phi}}{\gamma_{\omega}}\right) \left(\frac{U_{A\phi}}{U_A}\right) \cos(\phi') \approx \left(\cos^2(\phi) + \frac{1}{2}\sin^2(\phi)\right) \cos(\phi) = \frac{1}{2} \left(\cos(\phi) + \cos^3(\phi)\right) \quad (4.69)$$

Equation 4.69 can be further expanded in terms of ϕ and results in

$$\left(\frac{\gamma_{\omega\phi}}{\gamma_{\omega}}\right) \left(\frac{U_{A\phi}}{U_A}\right) \cos(\phi') \approx 1 - \phi^2 \quad (4.70)$$

Equation 4.70 is incidentally equal to the first two terms of the power series for the $\cos^2(\phi)$ function. Hence the curve of the $\cos^2(\phi)$ was compared with the calculated results in Equation 4.68 for three γ_{ω} cases. The calculated results show agreement with the curve of $\cos^2(\phi)$ in spite of the large γ_{ω} . Therefore, we adopted the formula of $\cos^2(\phi)$ to express the decreasing ratio of the horizontal component of the lift force in place of Equation 4.68.

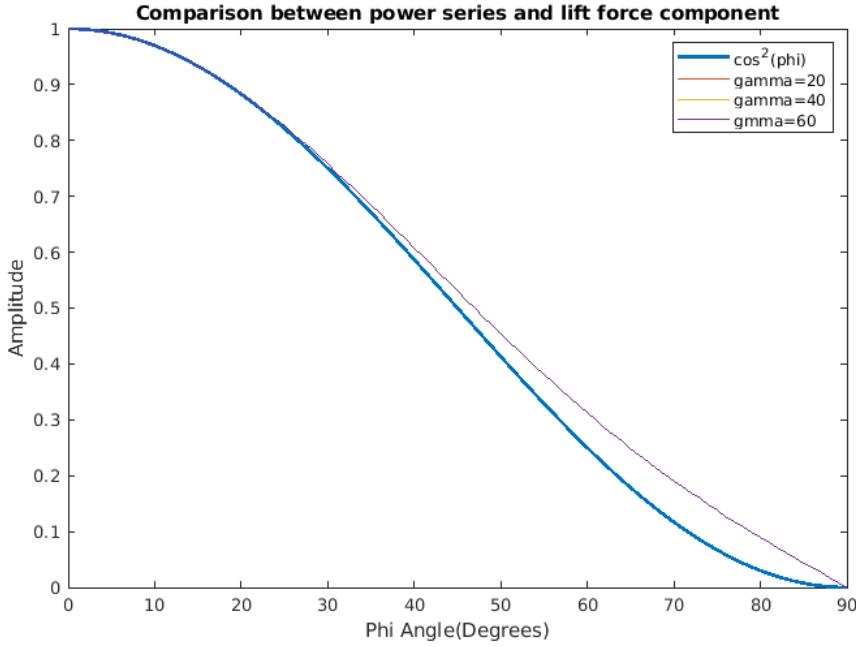


Figure 4.16: Lift force component represented as a power curve

Finally, when the lift coefficient represents the variation of the lift force including the contribution of dynamic pressure of apparent wind speed, the horizontal component of lift coefficient in the heeled condition L'_s is described as:

$$L'_s = L'_{s0} \cos^2(\phi) \quad (4.71)$$

The main part of the drag is caused by the induced drag, which is in proportion to the square of the lift force. The reduction of lift force expressed by Equation 4.66 is also approximated by $\cos(\phi)$. The vector of the drag force is in line with the apparent wind axis and does not incline by the heel angle. Therefore the horizontal component of the drag coefficient D'_s is described as:

$$D'_s = D'_{s0} \cos^2(\phi) \quad (4.72)$$

From these results, the aerodynamic coefficients in the horizontal components X'_s and Y'_s are then expressed as follows using the coefficients at the upright condition L'_{s0} and D'_{s0} :

$$X'_s = L'_s \sin(\gamma_\omega) - D'_s \cos(\gamma_\omega) = L'_{s0} \cos^2(\phi) \sin(\gamma_\omega) - D'_{s0} \cos^2(\phi) \cos(\gamma_\omega) = X'_{s0} \cos^2(\phi) \quad (4.73)$$

$$Y'_s = L'_s \cos(\gamma_\omega) - D'_s \sin(\gamma_\omega) = L'_{s0} \cos^2(\phi) \cos(\gamma_\omega) - D'_{s0} \cos^2(\phi) \sin(\gamma_\omega) = Y'_{s0} \cos^2(\phi) \quad (4.74)$$

The moment K_s is generated mainly by the Y_s force, however it is also affected by the component normal to the mast, hence

$$K'_s = -Y'_s \left(\frac{z_{GCE}^G}{\sqrt{S_A}} \right) / \cos(\phi) \quad (4.75)$$

where z_{GCE}^G is the z-coordinate of the geometric center of effort of the sail from the CG of the boat and negative upwards.

The moment N_s is also generated mainly by the Y_s force, however, it is well known that the N_s is also affected by the heel angle ϕ due to the application point of the thrust force X_s moving outboard to lee side. Therefore N'_s can be written, including the effect of X'_{s0} as

$$N'_s = \left(Y'_{s0} \frac{x_{GCE}^G}{\sqrt{S_A}} + X'_{s0} \frac{z_{GCE}^G}{\sqrt{S_A}} \sin(\phi) \right) \cos^2(\phi) \quad (4.76)$$

where x_{GCE}^G is x-coordinate of the geometric center of effort of the sail from CG of the boat.

The sail force and moment components defined above are shown below:

$$X_s = X'_{s0} \cos^2(\phi) \times \frac{1}{2} \rho_a U_A^2 S_A \quad (4.77)$$

$$Y_s = Y'_{s0} \cos^2(\phi) \times \frac{1}{2} \rho_a U_A^2 S_A \quad (4.78)$$

$$K'_s = -Y'_s \left(\frac{z_{GCE}^G}{\sqrt{S_A}} \right) / \cos(\phi) \times \frac{1}{2} \rho_a U_A^2 S_A \quad (4.79)$$

$$N'_s = \left(Y'_{s0} \frac{x_{GCE}^G}{\sqrt{S_A}} + X'_{s0} \frac{z_{GCE}^G}{\sqrt{S_A}} \sin(\phi) \right) \cos^2(\phi) \times \frac{1}{2} \rho_a U_A^2 S_A \quad (4.80)$$

4.7. Linearizations and Simplifications

The state space model associated with the first order Nomoto model is both observable and controllable. The state space model associated with the second order Nomoto model is also observable, however it is controllable only if the effective sway time constant is different from the effective yaw time constant. The first order model is very popular in adaptive autopilot applications. Model reduction for a fourth order transfer function ship model describing the sway-yaw-roll dynamics is conducted to reach the second order Nomoto model describing the sway-yaw dynamics and the first order Nomoto model describing the yaw dynamics itself.

Chapter 5

Sailing Control System Development

This chapter details the dynamic model reduction and linearizations that is then used to design the motion control systems of sailing. The chapter details a broad overview of the motion control system and goes into detail about the rudder and sail control, and also compares different proposed control strategies. It then proceeds to detail how a sailboat sails into the wind and the control system required to achieve this.

5.1. Sailing Control System Overview

The Sailing Control System (SCS) consists of a combination of different control systems, namely classical control, fuzzy logic control, gain scheduling and state machines. The reason for such a large combination of control systems is due to the varying maneuverability requirements required when sailing into the wind. Classical control is primarily used in controlling the rudder, as it has been shown in the Chapter ?? and in [26]. The fuzzy logic control and gain scheduling is used when sailing into the wind. The switch between forms of control a state-machine will be used.

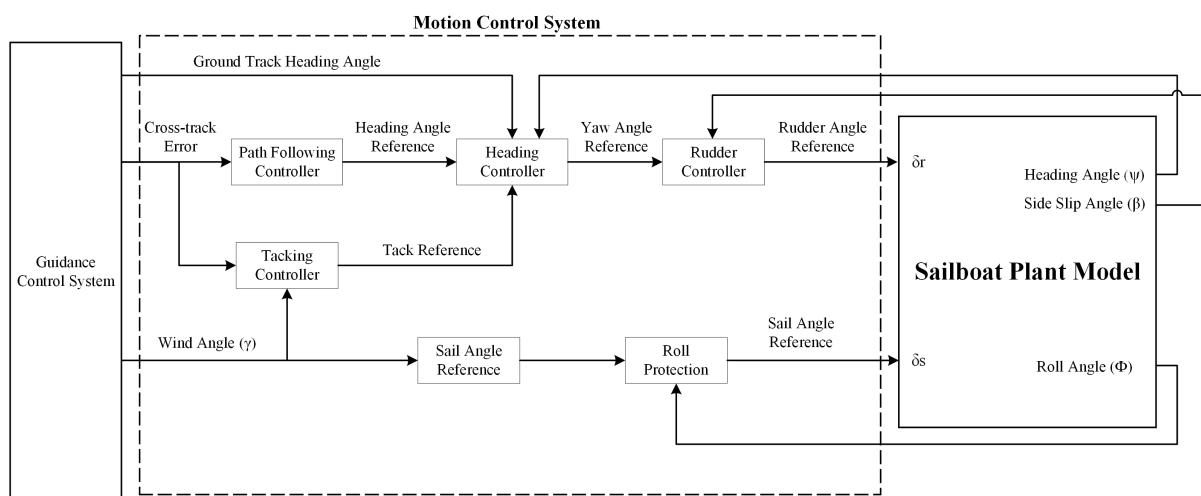


Figure 5.1: Block diagram overview of SCS

5.2. Autopilot Design

5.2.1. Ship Steering Dynamics Model Reduction

For ship autopilots design, a simple model with average predicting capability is usually preferred. Based on the linearized surge-sway-yaw-roll equations of motion, a fourth order transfer function relating the yaw rate to the rudder angle is derived.

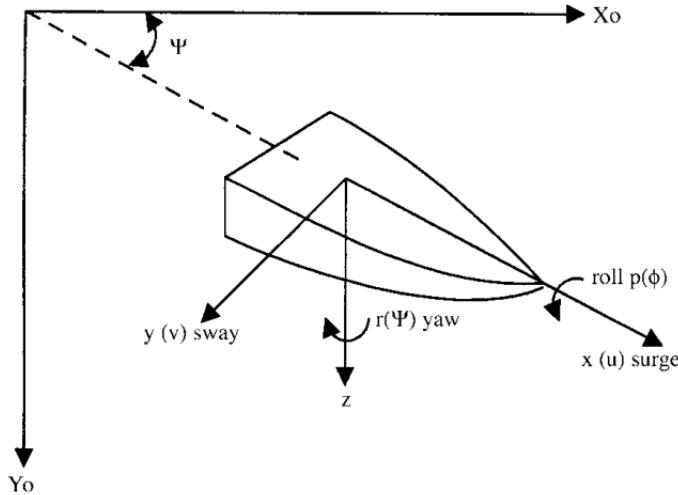


Figure 5.2: Sway-yaw-roll motion coordinate system

Upon linearization with respect to a straight line motion with a constant forward speed u_0 , the surge equation is decoupled and the following linear coupled sway-yaw-roll equations follow,

$$m(\dot{v} + u_0 r) = Y_v V + Y_{\dot{v}} \dot{V} + Y_\phi \phi + Y_p P + Y_{\dot{p}} \dot{P} + Y_r r + Y_{\dot{r}} \dot{r} + Y_\delta \delta \quad (5.1)$$

$$I_x \ddot{\phi} = K_v V + K_{\dot{v}} \dot{V} + Y_\phi \phi + K_p P + K_{\dot{p}} \dot{P} + K_r r + K_{\dot{r}} \dot{r} + K_\delta \delta - mgGM\phi \quad (5.2)$$

$$I_z \ddot{\psi} = N_v V + N_{\dot{v}} \dot{V} + N_\phi \phi + N_p P + N_{\dot{p}} \dot{P} + N_r r + N_{\dot{r}} \dot{r} + N_\delta \delta \quad (5.3)$$

where Y_v , $Y_{\dot{v}}$, ..., indicate the hydrodynamic coefficients; for instance Y_v indicates the derivative of the sway force Y to the sway speed V evaluated at the reference condition; m is the mass of the ship; I_x is the moment of inertia about the x-axis; I_z is the moment of inertia about the z-axis; V is the sway speed; u is the surge speed; r is the yaw rate; ψ is the heading angle defined by $\dot{\psi} = r$; p is the roll rate; ϕ is the roll angle defined by $\dot{\phi} = p$ and GM is the metacentric height, which indicates the restoring capability of a ship in rolling motion. Taking the Laplace transform of Equations 6.1-3 and rearranging, we have

$$a_1 V = a_2 \Phi + a_3 r + a_4 \delta \quad (5.4)$$

$$b_1 \Phi = b_2 V + b_3 r + b_4 \delta \quad (5.5)$$

$$c_1 r = c_2 V + c_3 \Phi + c_4 \delta \quad (5.6)$$

where

$$a_1 = m - Y_{\dot{v}} S - Y_v \quad (5.7)$$

$$a_2 = Y_{\dot{p}} S^2 + Y_p S + Y_\phi \quad (5.8)$$

$$a_3 = Y_{\dot{r}} S + Y_r + mu_0 \quad (5.9)$$

$$a_4 = Y_\delta \quad (5.10)$$

$$b_1 = (I_X - K_{\dot{p}}) S^2 - K_p S + mgGM \quad (5.11)$$

$$b_2 = K_{\dot{v}} S + K_v \quad (5.12)$$

$$b_3 = K_{\dot{r}} S + K_r \quad (5.13)$$

$$b_4 = K_\delta \quad (5.14)$$

$$c_1 = (I_Z - N_{\dot{r}}) S - N_r \quad (5.15)$$

$$c_2 = N_{\dot{v}} S + N_v \quad (5.16)$$

$$c_3 = N_{\dot{p}} S^2 + N_p S + N_\phi \quad (5.17)$$

$$c_4 = N_\delta \quad (5.18)$$

After eliminating the sway speed V and roll angle ϕ from Equations 6.4-6, the following transfer function relating the yaw rate r to the rudder angle δ can be obtained:

$$\frac{r}{\delta} = \frac{a_1(b_1 c_4 + b_4 c_3) + a_2(b_4 c_2 - b_2 c_4) + a_4(b_1 c_2 + b_2 c_3)}{a_1(b_1 c_1 - b_3 c_3) - a_2(b_2 c_1 + b_3 c_2) - a_3(b_1 c_2 + b_2 c_3)} \quad (5.19)$$

It can be easily verified that the numerator of Equation 5.19 is third order in S , while the denominator is fourth order in S . Hence, Equation 5.19 can be expressed in the following form.

$$\frac{r}{\delta} = \frac{K(1 + T_3 S)(S^2 + 2n\omega_0 S + w_0^2)}{(1 + T_1 S)(1 + T_2 S)(S^2 + 2\delta\omega_n S + w_n^2)} \quad (5.20)$$

where the quadratic factors are due to the coupling effect from the roll mode on the yaw rate. The zero $(1 + T_3S)$ and the pole $(1 + T_2S)$ are due to the coupling effect from the sway mode on the yaw dynamics. If the roll mode is neglected, Equation 5.20 can be further reduced to the following form

$$\frac{r}{\delta} = \frac{K(1 + T_3S)}{(1 + T_1S)(1 + T_2S)} \quad (5.21)$$

Equation 5.21 is known as the second order Nomoto model, where K is the static yaw rate gain, and T_1 , T_2 and T_3 are time constants. Numerical values of the parameters in Equation 5.21 for a Mariner class vessel are given by $T_1 = 118$, $T_2 = 7.8$, $T_3 = 18.5$ and $K = 0.185$. The zero term $(1 + T_3S)$ and the high frequency pole term $(1 + T_2S)$ are due to the coupling effect from the sway mode. In practice, because the pole term $(1 + T_2S)$ and the zero term $(1 + T_3S)$ in Equation 5.21 nearly cancel each other, a further simplification can be done to give first order Nomoto model

$$\frac{r}{\delta} = \frac{K}{(1 + TS)} \quad (5.22)$$

where

$$T = T_1 + T_2 - T_3 \quad (5.23)$$

in Equation 5.23 is called the effective yaw rate time constant. It is obtained by equating the right hand side of Equation 5.21 to the right hand side of Equation 5.22 requiring the equality relationship to hold up to first order in S . If $T_2 = T_3$; namely, a perfect cancellation occurs, then the equality relationship is true up to second order in S , then T is of course, equal to T_1 . For the Mariner class vessel, the value of the effective constant T in Equation 5.23 is given by $T = 107.3$.

The first order Nomoto model defined by Equation 5.22 is widely employed in the ship steering autopilot design. The yaw dynamics is characterized by the parameters K and T , which can be easily identified from standard maneuvering tests. In practice, ship steering autopilots are designed for heading angle control. Hence, it is the transfer function relating the heading angle ψ to the rudder angle δ being needed in the autopilot design. Since the yaw rate r is actually the time derivative of ψ , the required transfer function can be readily obtained by adding an integrator $\frac{1}{s}$ to the transfer function models defined by Equation 5.21 and Equation 5.22.

5.2.2. First Order Nomoto Model

The controllability, observability and identifiability properties of the first order Nomoto model-based system will be discussed in this section. The identifiability property will be

discussed with respect to the first order Nomoto transfer function model. However, the controllability and observability properties have to be discussed with respect to the state-space model derived from the Nomoto model. This is because the transfer function model always represents the observable and controllable parts of the system dynamics. If there is any unobservable or uncontrollable parts of the dynamics, they are cancelled out before reaching the transfer function model form. Hence, it only makes sense to discuss the controllability and observability properties of the state space model, which contains the observable, controllable and uncontrollable modes. Equation 5.22 can be expressed in time domain as

$$T\dot{r} + r = K\delta \quad (5.24)$$

with the notation

$$\dot{\psi} = r \quad (5.25)$$

where ψ is the heading of the ship. Equation 5.24 can be written as

$$T\ddot{\psi} + \dot{\psi} = K\delta \quad (5.26)$$

Equation 5.25 and 5.26 can be arranged in the standard state space form

$$\dot{x} = Ax + Bu \quad (5.27)$$

$$y = Cx + du \quad (5.28)$$

where

$$x = \begin{bmatrix} \psi \\ r \end{bmatrix} \quad (5.29)$$

$$u = \delta \quad (5.30)$$

$$y = \psi \quad (5.31)$$

and

$$A = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{1}{T} \end{bmatrix} \quad (5.32)$$

$$B = \begin{bmatrix} 0 \\ \frac{K}{T} \end{bmatrix} \quad (5.33)$$

$$C = \begin{bmatrix} 1 & 0 \end{bmatrix} \quad (5.34)$$

The first order system is both controllable and observable. Consequently, the first order

Nomoto model satisfies the identifiability property, hence, on-line estimation of the model parameters based on the measured rudder and yaw rate information will be possible and adaptive control strategy can be successfully implemented.

The first order Nomoto model being very popular in the design for ship steering autopilots is not without reason. Its simplicity and reasonable accuracy in describing small rudder angle yaw dynamics make it attractive. Moreover, the relative easiness in identifying the model parameters makes it suitable for adaptive autopilot application, where on-line estimation of the model parameters is important. The second order Nomoto model includes the coupling effect from the sway to the yaw mode. This introduces a zero and a high frequency pole into the transfer function. The zero structure contributes to the overshoot phenomenon, which can be seen in the yaw rate for large rudder angle maneuvers. However, the ill-conditioning problem associated with the second order Nomoto model in the identification of the model parameters from input-output data seems to outweigh the improvement gained in its modeling capability. An approach that retains the zero structure while avoiding the ill-conditioning problem during on-line estimation has been proposed. The bode plots of the first, second and fourth order model is shown below,

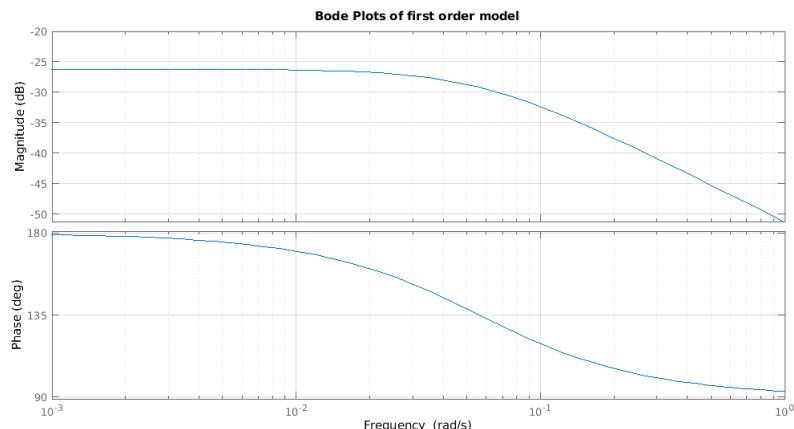
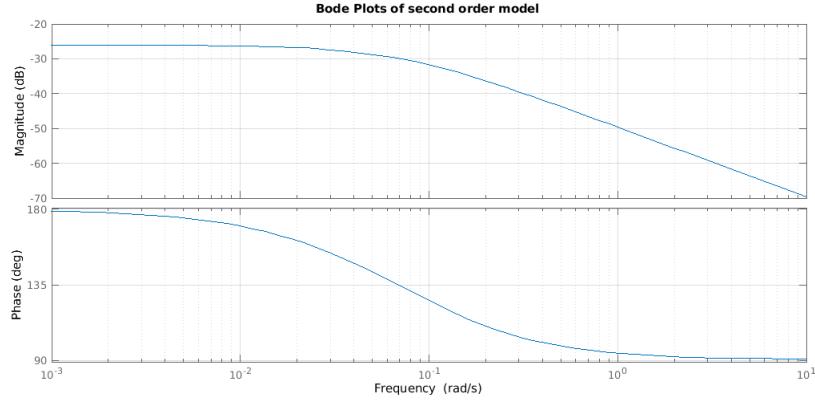
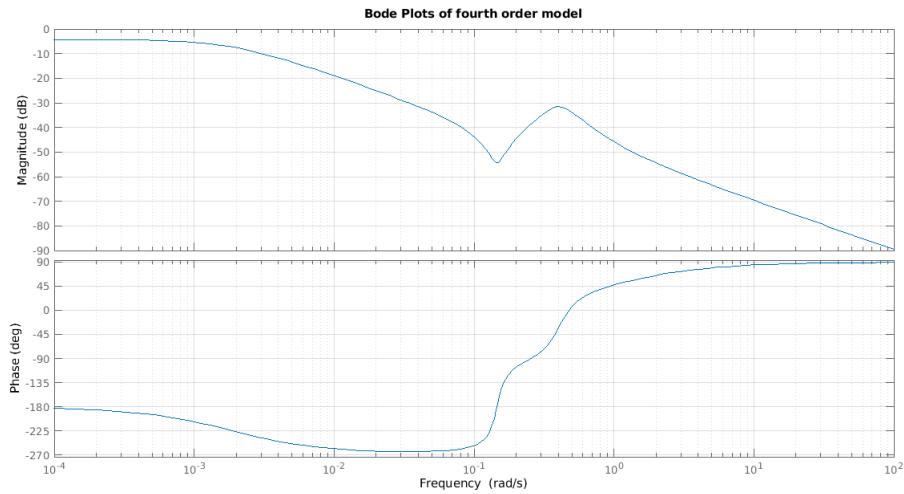


Figure 5.3: First Order bode Plot

**Figure 5.4:** Second Order Bode Plot**Figure 5.5:** Fourth Order Bode Plot

5.2.3. Stabilization

Conventional PID control systems have their origin from SISO linear systems theory. However, it is possible to generalize this to nonlinear MIMO systems by using results from robotics. This requires that the ocean vessel equations of motion are expressed in a vectorial setting:

$$\dot{n} = J_\Theta(n)v \quad (5.35)$$

$$M\dot{v} + C(v)v + D(v)v + g(n) = \tau + \omega \quad (5.36)$$

For this model class, MIMO nonlinear PID control systems can be designed by exploiting the fact that the mass matrix is positive definite and constant ($M = M^T > 0$, $\dot{M} = 0$),

the Coriolis and centripetal matrix $C(v) = -C^T(v)$ is skew-symmetrical and the damping matrix $D(v) > 0$ is strictly positive.

Open-Loop Stability and Maneuverability

When designing a motion control system a compromise between stability and maneuverability must be made,

- *Stability* of an uncontrolled ocean vessel can be defined as the ability to return to an equilibrium point after a disturbance, without any corrective action of the actuators
- *Maneuverability*, on the other hand, is defined as the capability of the craft to carry out specific maneuvers.

It is well known that a craft that is easy to maneuver, for instance fighter aircraft or a high-speed watercraft, can be marginally stable, or even unstable in open loop. On the other hand, excessive stability implies that the control effort will be excessive in a maneuvering situation whereas a marginally stable ship is easy to maneuver.

Consequently, a compromise between stability and maneuverability must be made.

Straight-Line, Directional and Positional Motion Stability

For ocean vessels it is common to distinguish between three type of stability, namely:

- Straight-line stability
- Directional or course stability
- Positional motion stability

This can be explained using open-loop and closed-loop stability analyzes. In order to understand the different types of stability one can consider the following test system:

$$\dot{x} = u\cos(\psi) - v\sin(\psi) \approx u_0\cos(\psi) \quad (5.37)$$

$$\dot{y} = u\sin(\psi) + v\cos(\psi) \approx u_0\sin(\psi) \quad (5.38)$$

$$\dot{\psi} = r \quad (5.39)$$

$$T\dot{r} + r = K\delta + \omega \quad (5.40)$$

where ω is the external disturbances and $u_0 = \text{constant}$ is the cruise speed. The first two equations represent the (x, y) position of the vessel while the last two equations describe

the yaw dynamics modeled by Nomoto's first-order model. For simplicity, it is assumed that the yaw motion of the vessel is stabilized by a PD-controlled rudder servo:

$$\delta = -K_p(\psi - \psi_d) - K_d r \quad (5.41)$$

where $\psi_d = \text{constant}$ denotes the desired heading angle and K_p and K_d are two positive regulator gains. Substituting the control law, in Equation 5.41, into Nomoto's first-order model, in Equation 5.39, yields the closed-loop system

$$T\ddot{\psi} + (1 + KK_d)\dot{\psi} + KK_p\psi = KK_p\psi_d + \omega \quad (5.42)$$

The closed-loop system represents a second-order mass-damper-spring system

$$m\ddot{\psi} + d\dot{\psi} + k\psi = f(t) \quad (5.43)$$

with driving input

$$f(t) = k\psi_d + \omega \quad (5.44)$$

The eigenvalues $\lambda_{1,2}$, the natural frequency w_n and the relative damping ratio ξ for the mass-damper-spring system are

$$\lambda_{1,2} = \frac{-d \pm \sqrt{d^2 - 4km}}{2m} \quad (5.45)$$

$$\omega_n = \sqrt{\frac{k}{m}} \quad (5.46)$$

$$\xi = \frac{d}{2}\frac{1}{\sqrt{km}} \quad (5.47)$$

Instability: For uncontrolled ocean vessel ($K_p = K_d = 0$) instability occurs when

$$\lambda_1 = -\frac{d}{m} = -\frac{1}{T} > 0 \quad (5.48)$$

$$\lambda_2 = 0 \quad (5.49)$$

which simply states that $T < 0$.

Straight-Line Stability: Consider an uncontrolled ocean vessel ($K_p = K_d = 0$) moving in a straight path. If the new path is straight after a disturbance ω in yaw the craft is said to have straight-line stability. The direction of the new path will usually differ from the initial path because no restoring forces are present ($k=0$). This corresponds to

$$\lambda_1 = -\frac{d}{m} = -\frac{1}{T} < 0 \quad (5.50)$$

$$\lambda_2 = 0 \quad (5.51)$$

Consequently, the requirement $T>0$ implies straight-line stability for the uncontrolled craft ($\xi=0$).

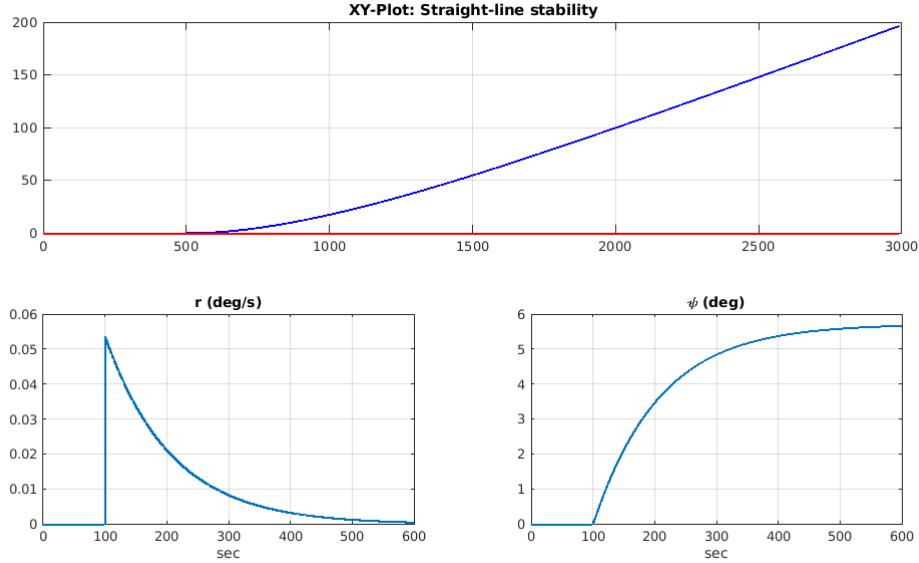


Figure 5.6: Straight Line stability

Directional Stability(Stability on Course): Directional stability is a much stronger requirement than straight-line stability. Directional stability requires the final path to be parallel to the initial path that is obtained $K_p > 0 \Rightarrow k > 0$. Additional damping is added through $K_d > 0$. This corresponds to PD control. A ocean vessel is said to be directional stable if both eigenvalues have negative real parts, that is

$$Re\{\lambda_{1,2}\} < 0 \quad (5.52)$$

The following two types of directional stability are observed: *No oscillations* ($d^2 - 4km \leq 0$): This implies that both eigenvalues are negative and real, that is $\xi \leq 1$ such that

$$\lambda_{1,2} = \frac{-d \pm \sqrt{d^2 - 4km}}{2m} = \left(-\xi \pm \sqrt{\xi^2 - 1}\right) \omega_n < 0 \quad (5.53)$$

For a critically damped system $\xi = 1$, such that $\lambda_{1,2} = -1/2(d/m) = -\omega_n$.

Damped Oscillator ($d^2 - 4km < 0$): This corresponds to two imaginary eigenvalues $\lambda_{1,2}$, with negative real parts ($\xi < 1$), that is

$$\lambda_{1,2} = \frac{-d \pm j\sqrt{d^2 - 4km}}{2m} = \left(-\xi \pm j\sqrt{\xi^2 - 1} \right) \omega_n \quad (5.54)$$

Directional stability for a critically damped and under damped ocean vessel is illustrated in Figure 5.7 and 5.8. Notice the oscillations in both positions and yaw angle in underdamped ocean vessel. Directional stability requires feedback control since there are no restoring forces in yaw. However in heave, roll and pitch where metacentric restoring forces are present ($k > 0$) no feedback is required to damp out the oscillations.

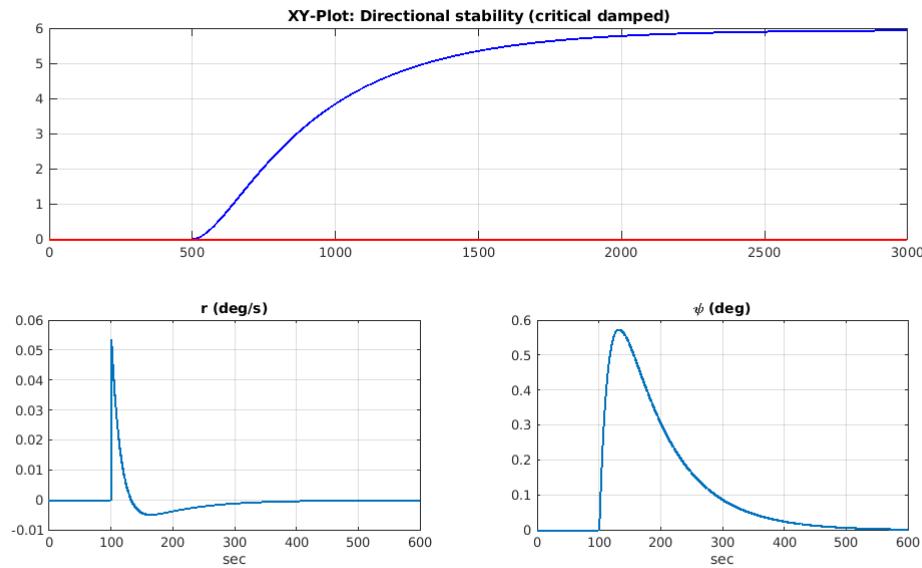


Figure 5.7: Directional Stability(critically damped)

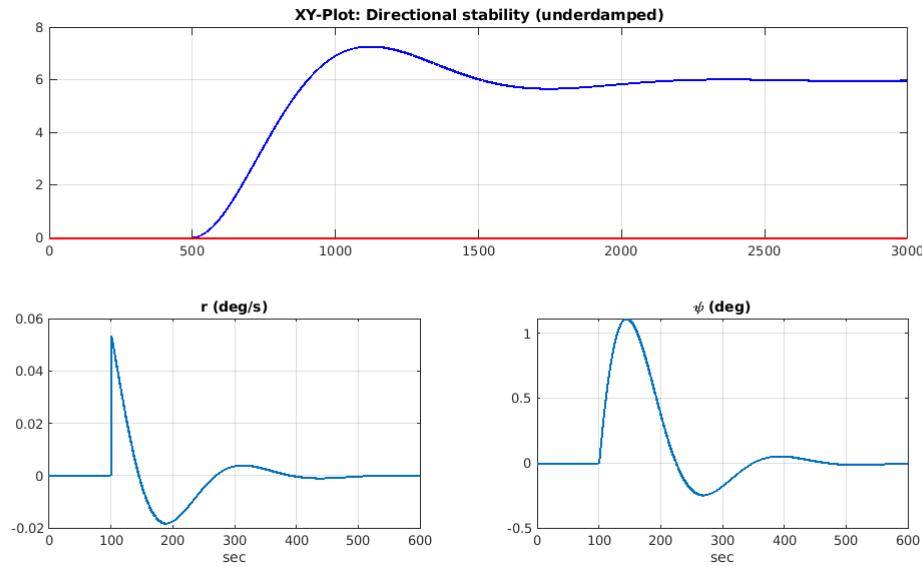


Figure 5.8: Directional Stability(under damped)

Positional Motion Stability: Positional motion stability implies that the ship should

return to its original path after a disturbance, illustrated in Figure 5.9. This can be achieved by including the integral action in the controller. Hence, a PID controller can be designed to compensate for the unknown disturbance term ω while a PD controller will generally result in a steady-state offset.

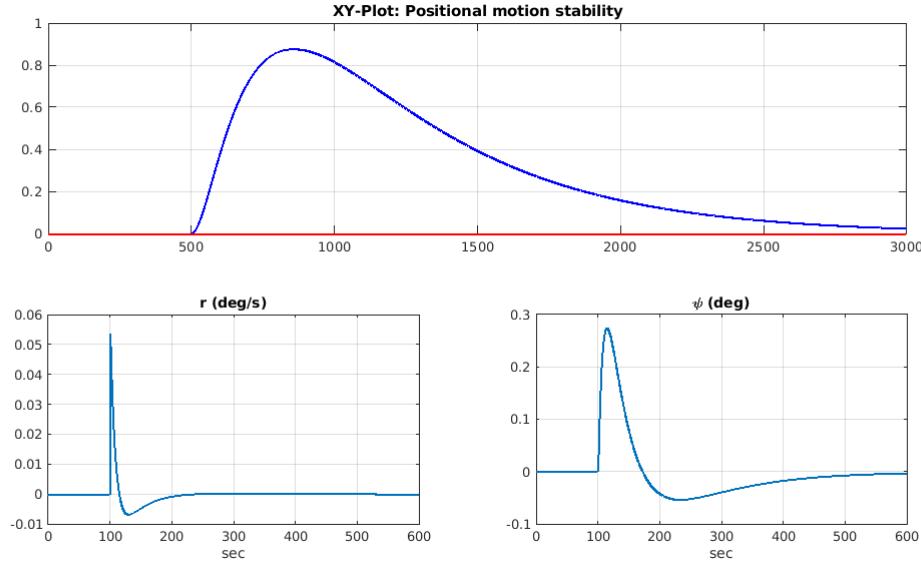


Figure 5.9: Positional Motion Stability

The proposed PID controller is illustrated in Figure 5.10. The input to the PID controller is illustrated as V_{ref} , which will be determined by the heading controller. The output of the PID controller will be the rudder angle. K_i , K_p , K_d is the respective gains of the integrator, proportional and derivative parts.

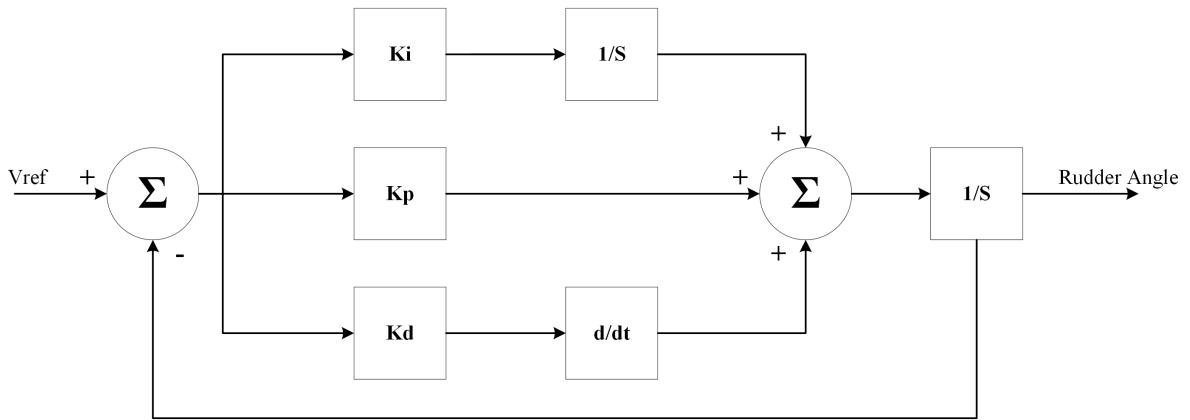


Figure 5.10: PID rudder controller

The control law for the rudder angle can thus be written as,

$$\delta_r = K_p e + K_I \int_0^t e(\tau) d\tau + K_D \frac{de}{dx} \quad (5.55)$$

5.2.4. Acceleration Feedback

It is possible to extend the results obtained for the use of a PID to include acceleration feedback. As shown in Equation 5.43, that the steering dynamics can be presented as a mass-damper system. The system is expanded to include acceleration feedback shown below,

$$m\ddot{\psi} + d\dot{\psi} + k\psi = \tau_{PID} - K_m\ddot{\psi} + \omega \quad (5.56)$$

where $K_m > 0$ is the acceleration feedback gain and τ_{PID} represents the PID controller. This results in

$$(m + K_m)\ddot{\psi} + d\dot{\psi} + k\psi = \tau_{PID} + \omega \quad (5.57)$$

From the above equation it is noticed that the acceleration feedback increases the mass from m to $m + K_m$ and also reduces the gain in front of the disturbances ω from $1/m$ to $1/(m + K_m)$. This results in a system being less sensitive to an external disturbance ω if acceleration feedback is applied. The design can be further improved by introducing a frequency-dependent virtual mass in the following form,

$$\tau = \tau_{PID} - h_m(s)\ddot{x} \quad (5.58)$$

If $h_m(s)$ is chosen as a low-pass filter,

$$h_m(s) = \frac{K_m}{T_m s + 1} \quad (5.59)$$

with the gain $K_m > 0$ and the time constant $T_m > 0$, the new control law would be,

$$\left(m + \frac{K_m}{T_m s + 1} \right) \ddot{\psi} + d\dot{\psi} + k\psi = \tau_{PID} + \omega \quad (5.60)$$

5.2.5. Proposed other Control Techniques

Another control law, proposed and tested in [3], for the rudder angle is using a non-linear proportional gain depending on the heading error shown below,

$$k(e) = \frac{k_p}{1 + c_p|e|} \quad (5.61)$$

Combining the two control laws results in the following non-linear rudder control that still maintains all the motion stability.

$$\delta_r = \frac{k_p}{1 + c_p|e|} e + K_I \int_0^t e(\tau) d\tau + K_D \frac{de}{dx} \quad (5.62)$$

The rudder reference angle is illustrated in Figure 5.11.

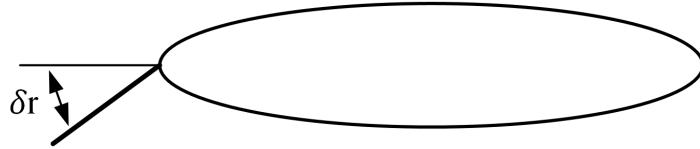


Figure 5.11: Rudder reference angle

5.3. Sail Controller Design

To design a sail controller a simplified version of propulsion control, defined in [27], is looked at.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} u \cos(\psi) - v \sin(\psi)\cos(\phi) \\ u \sin(\psi) - v \cos(\psi)\cos(\phi) \end{bmatrix} \quad (5.63)$$

The force generated by the sail is shown in Equation 5.64 with $f_s = ||F_s||$.

$$F_s = \begin{bmatrix} f_s \sin(\delta_s) \\ -f_s \cos(\delta_s) \cos(\phi) \end{bmatrix} \quad (5.64)$$

The angle of attack on the sail is determined by the direction of the apparent wind vector and the sail angle, which is

$$AoA = \pi - (\delta_s - \psi_{aw}) \quad (5.65)$$

The force of the sail discussed in Section 4.6 can be simplified in the surge direction as

$$f_s = -Kv_{aw}^2 \sin(\delta_s - \psi_{aw}) \sin(\delta_s) \quad (5.66)$$

where K is dependent on the sail dimensions, v_{aw} is the apparent wind speed, δ_s is the sail angle and ψ_{aw} is the apparent wind angle. The sail reference angle is illustrated in Figure 5.12.

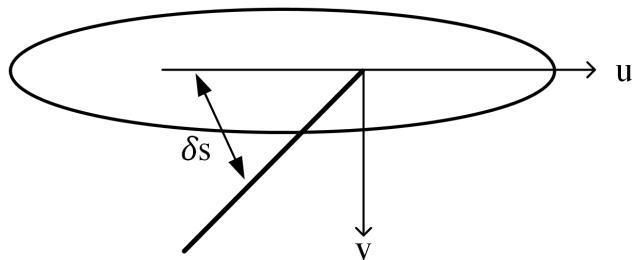


Figure 5.12: Sail Reference Angle

The sail control is usually done by adjusting the sail angle to the apparent wind angle with an offset. This is very simple and will result in a forward speed which as a bare minimum is sufficient. More extreme methods of sail control, developed in [28], is known as the extreme search method. This method consists of actively searching for the optimal sail angle by measuring the speed of the sailboat. When the sailboat speed is increasing the sail will be hauled in and when the sailboat's speed decreases the sail will be released. This method assumes that $\delta_s(t)$, which is the optimal sail angle is close to the current/base sail angle.

The new sail angle is updated as follows,

$$\delta_s(t) = \delta_s^b(t) + \alpha_u \quad (5.67)$$

where $\delta_s^b(t)$ is the current/base sail angle and α_u is the sail adjustment term defined as,

$$\alpha_u = K_p e_u(t) + K_i \int_0^t e_u(\tau) d\tau \quad (5.68)$$

where $e_u(t)$ is the error between desired and measured speed defined as,

$$e_u(t) = u_{max} - u(t) \quad (5.69)$$

A simple and very effective method of controlling the sail is the adjust the sail angle linearly accordingly to the apparent wind angle. The linear control is used to always achieved the optimal sail angle that will produced the most force in the surge direction. In Figure 5.13 the optimal sail angle is plotted against the apparent wind angle.

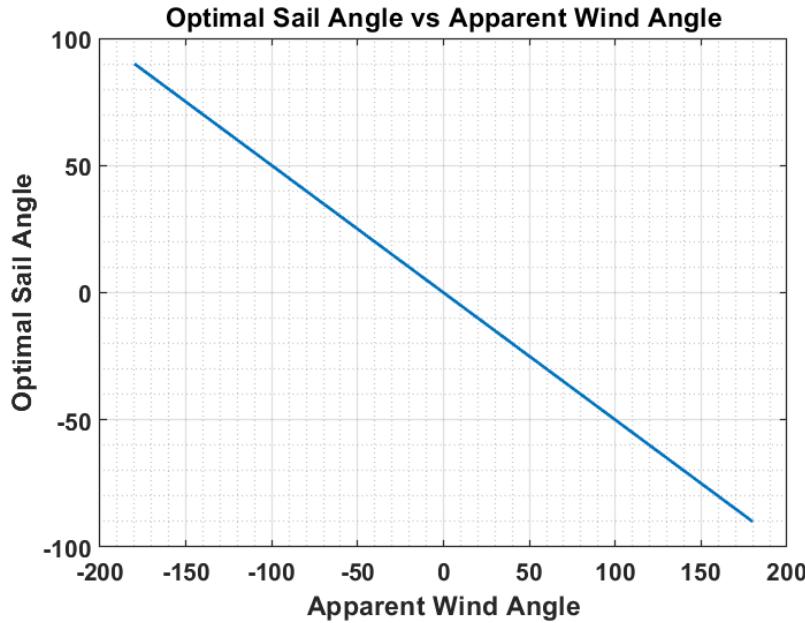


Figure 5.13: Optimal sail angle for varying wind angle

The sail angle be linearly adjusted by the following line equation,

$$\delta_s = -0.5 \times \psi_{aw} \quad (5.70)$$

with the constraints that the sail angle is limited to $-90 < \delta_s < 90$ and $-180 < \psi_{aw} < 180$. The dimensionless sail force is illustrated in Figure 5.14, the linear approximation of the optimal sail produces a sail force greater or equal than zero in the whole range of apparent wind angle.

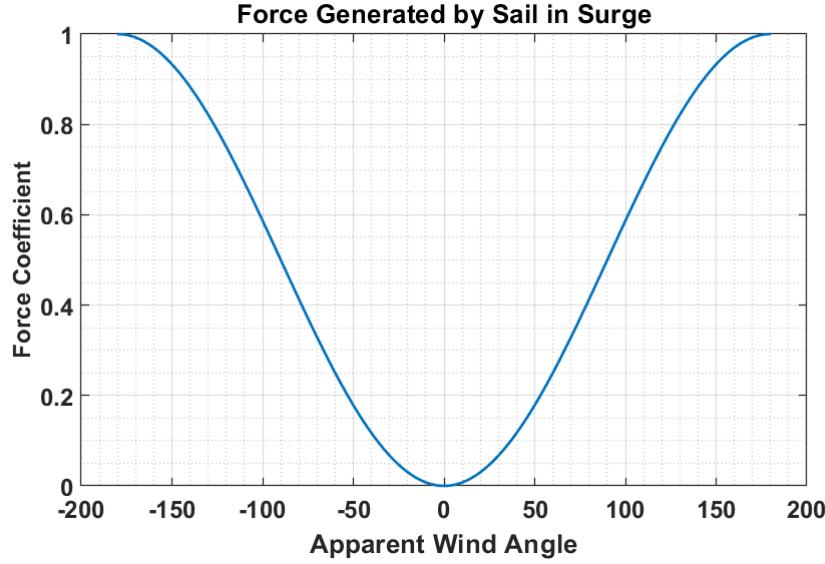


Figure 5.14: Dimensionless sail force

Although the extreme search method allows for better control over the sailboats speed it requires a lot more actuation of the sail.

5.4. Guidance and Waypoint Navigation

5.4.1. Guidance System

The guidance system is similar to the aircraft guidance system, discussed in [29]. The defined guidance consists of a series of straight-line path segments between waypoints. Each waypoint is a set of North and East coordinates on a map. The straight line between two consecutive waypoints is defined as the ground track. The purpose of the guidance controller is to control the ocean vessel onto the ground track by controlling the cross-track position error to zero. Given that you have the source waypoint and the destination waypoint, the heading angle and the length of the ground track can be calculated as illustrated in Figure

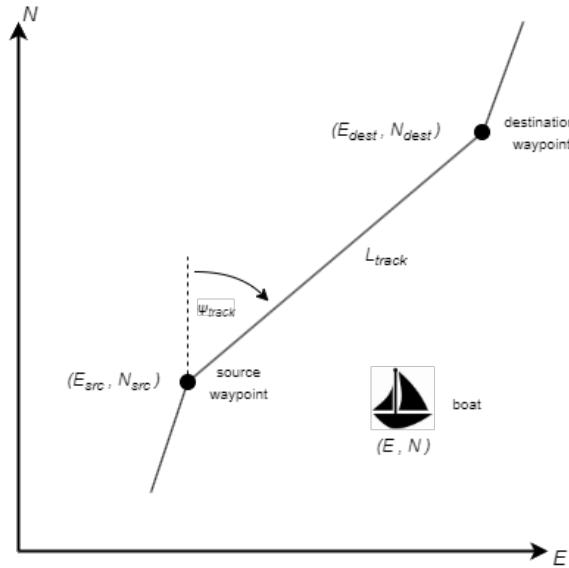


Figure 5.15: Ground track between source waypoint and destination waypoint

The track heading ψ_{track} and the track length L_{track} is calculated as

$$\psi_{track} = \tan^{-1} \left(\frac{E_{dest} - E_{src}}{N_{dest} - N_{src}} \right) \quad (5.71)$$

$$l_{track} = \sqrt{(N_{dest} - N_{src})^2 + (E_{dest} - E_{src})^2} \quad (5.72)$$

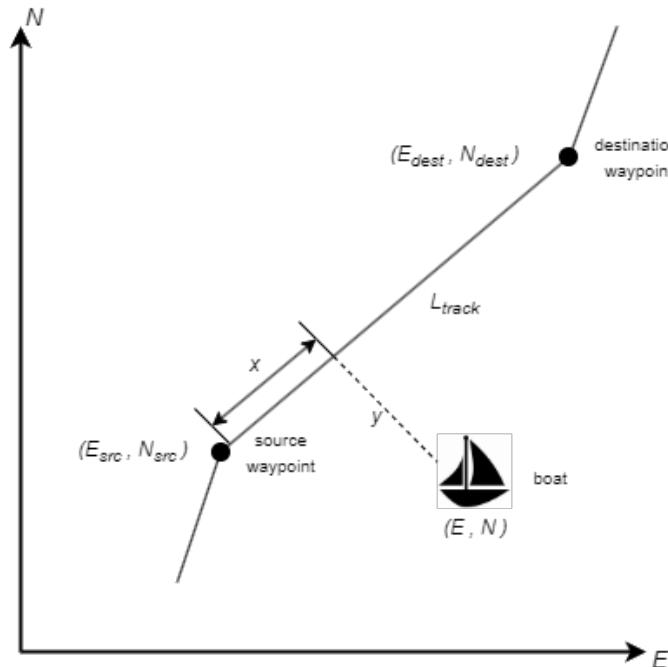


Figure 5.16: Cross-track error and in-track distance along track

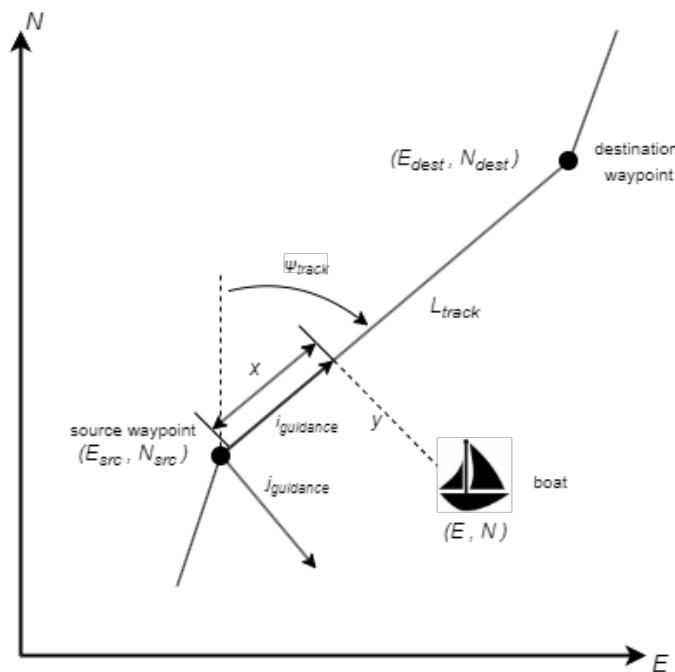


Figure 5.17: Guidance Axis System

The origin of the guidance axis system is at the location of the source waypoint, the x-axis is parallel to the ground track and pointing in the direction of the destination waypoint, its y-axis is perpendicular to the ground track, and its z-axis coincides with the down axis of the NED axis system. The guidance axis system is obtained by rotating the NED axis system through the track heading ψ_{track} , and by moving its origin to the location of the source waypoint. To obtain the cross-track error and the in-track distance, the boat position is first transformed from the NED axes to the guidance axes. The cross-track error is then simply the y-component in the guidance axis system, and the in-track distance is simply the x-component in the guidance axis system. The boat's position is transformed from the NED axis system to the guidance axis system with the following equation

$$\begin{bmatrix} s \\ e \end{bmatrix} = \begin{bmatrix} \cos(\psi_{track}) & \sin(\psi_{track}) \\ -\sin(\psi_{track}) & \cos(\psi_{track}) \end{bmatrix} \begin{bmatrix} N - N_{src} \\ E - E_{src} \end{bmatrix} \quad (5.73)$$

The boat is following the ground track when the boat heading equals the ground track heading and the cross-track error is equal to zero. The boat will have reached the destination waypoint when its in-track distance equals the length of the ground track.

5.4.2. Straight path Generation based on Waypoints

For surface craft vessels, discussed in [15], only two coordinates (x_k, y_k) for $k = 1, \dots, n$. The waypoint database therefore consists of

$$wpt.pos = (x_0, y_0), (x_1, y_1), \dots, (x_n, y_n) \quad (5.74)$$

Additionally other waypoint properties such as speed and heading is defined as,

$$wpt.speed = U_0, U_1, \dots, U_n \quad (5.75)$$

$$wpt.heading = \psi_0, \psi_1, \dots, \psi_n \quad (5.76)$$

In the case of a sailboat the speed is dependent on the wind and therefor is not considered in the waypoint navigation. The states that are thus important for sailboats during navigation are (x_i, y_i, ψ_i) , which are called the *pose*. The heading angle is usually not important but in the case of a sailboat it is important due to the constraint on sailing angles especially when performing the tacking maneuvering, which is discussed in Chapter

In 1957 Dubins [30] found the shortest path for path following and can be summarised as, "The shortest path (minimum time) between two configurations (x, y, ψ) of a craft moving

at constant speed U is a path formed by a straight lines and circular arc segments.” Since a craft is considered the start and end configurations are expressed in terms of positions (x, y) , heading angle ψ and in addition it is assumed there is bounds on the turning rate r . Although this method has drawbacks with a jump in turn rate, the use of arcs and straight lines is very simplistic. Figure 5.18 illustrates a path generated by the Dubins method. Also that needs to be considered is that the operator can specify a circle with radius R_i about each waypoint and can be stored in the database as

$$wpt.radius = R_0, R_1, \dots, R_n \quad (5.77)$$

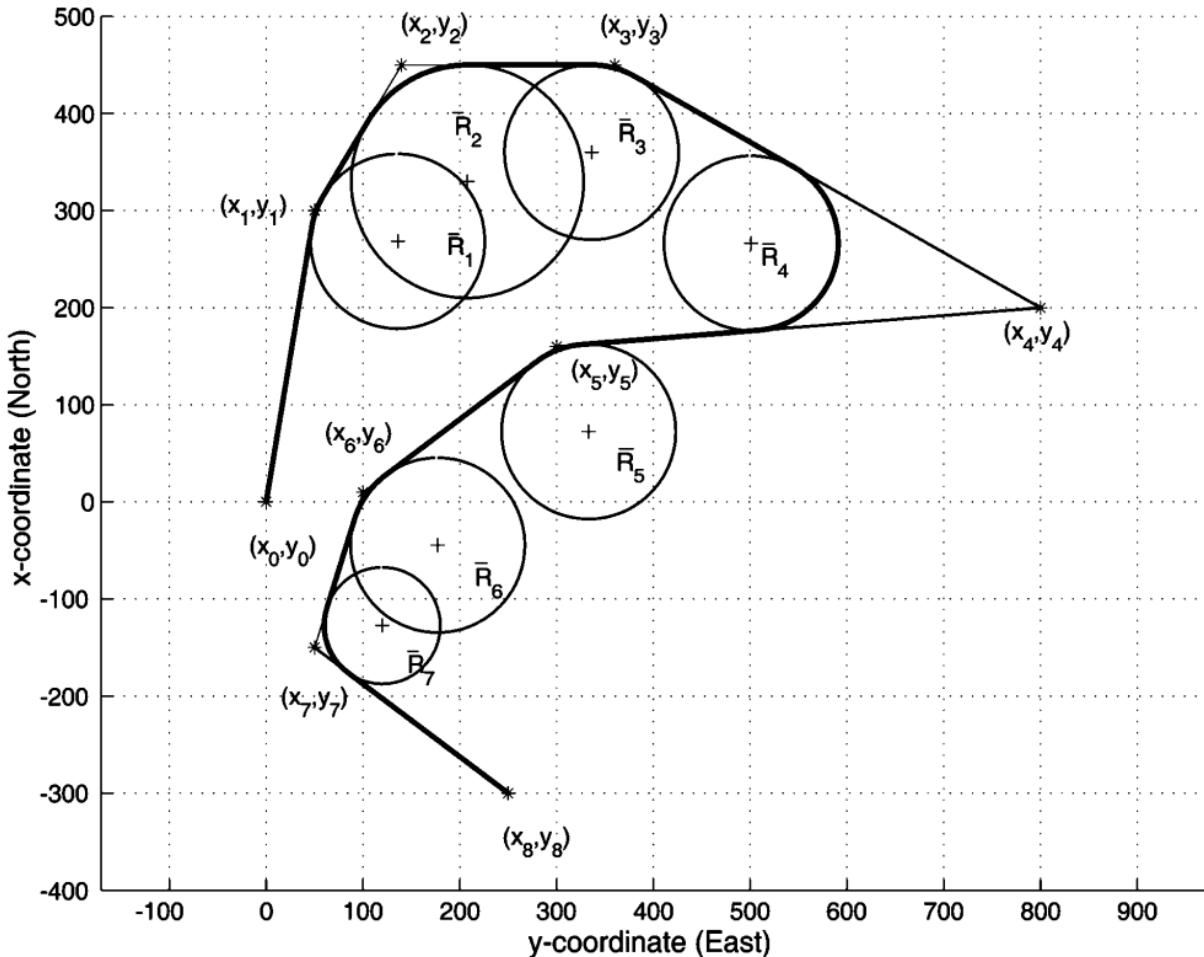


Figure 5.18: Straight lines and circular arc segments for waypoint guidance

5.4.3. Line-of-Sight (LOS) Steering Law

In order to steer the boat on a path a method known as the *lookahead – based steering* [31]. For path-following purposes, only the cross-track error is relevant since $e(t) = 0$ means that the boat has converged to the straight line. Thus the associated control objective for straight-line path following becomes

$$\lim_{t \rightarrow \infty} e(t) = 0 \quad (5.78)$$

For lookahead-based steering, the course angle assignment is separated into two parts:

$$x_d(e) = x_p + x_r(e) \quad (5.79)$$

where

$$x_p = \alpha_k \quad (5.80)$$

is the *path – tangential angle* illustrated in Figure 5.19, while

$$x_r(e) = \arctan\left(\frac{-e}{\Delta}\right) \quad (5.81)$$

is a *velocity-path relative angle*, which ensures that the velocity is directed toward a point on the path that is located a *lookahead distance* $\Delta(t) > 0$ ahead of the direct projection of $p^n(t)$ on to the path.

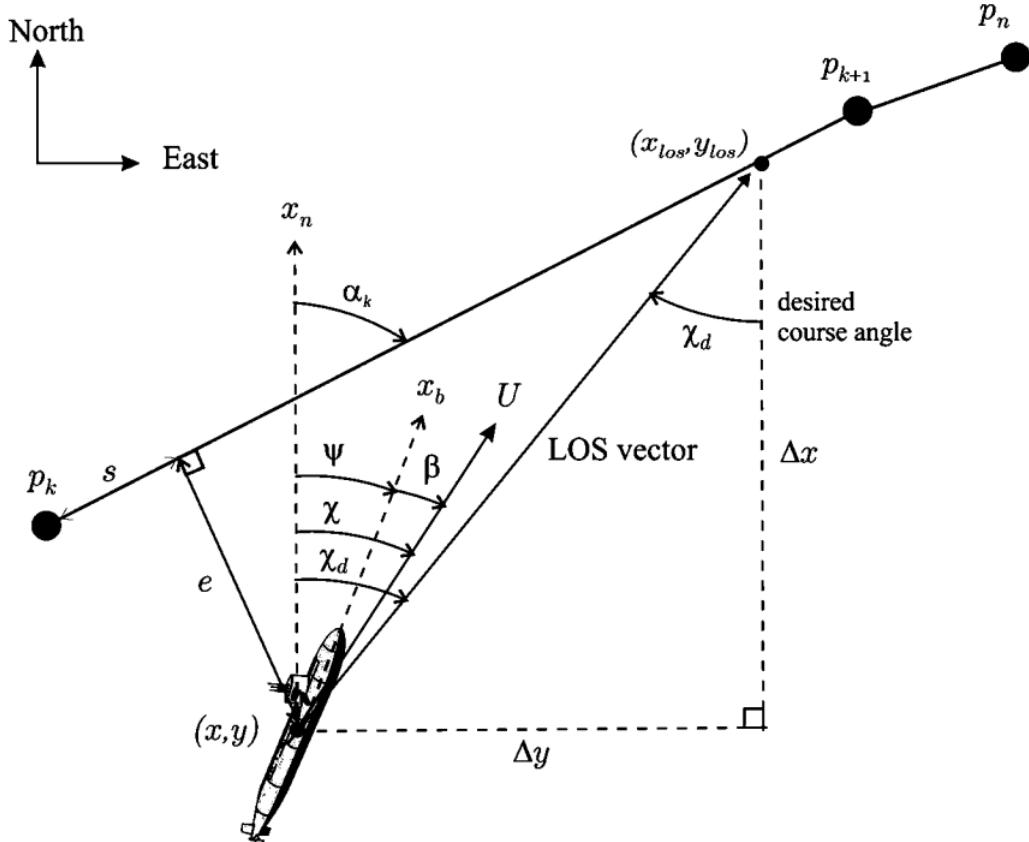


Figure 5.19: LOS guidance where the desired course angle x_d is chosen to point toward the LOS intersection point (x_{los}, y_{los})

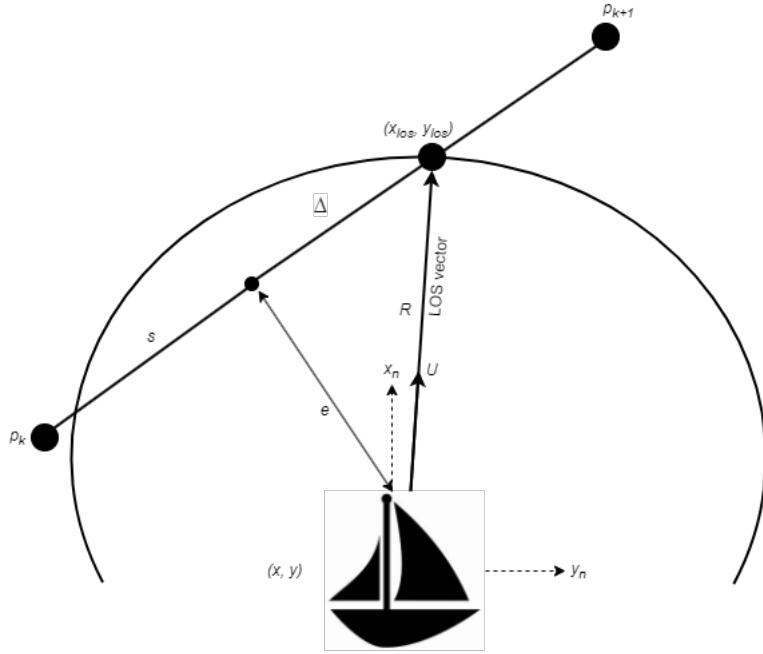


Figure 5.20: Circle of acceptance with constant radius R

Figure 5.20 illustrates that the circle of acceptance radius R is equal to

$$e(t)^2 + \Delta(t)^2 = R^2 \quad (5.82)$$

with

$$\Delta(t) = \sqrt{R^2 - e(t)^2} \quad (5.83)$$

varying between 0 and R for $|e(t)| = R$ and $|e(t)| = 0$, respectively. The steering law, in Equation 5.81, can also be interpreted as a saturating control law,

$$x_r(e) = \arctan(-K_p e) \quad (5.84)$$

where $K_p(t) = 1/\Delta(t) > 0$. Notice that the lookahead-based steering law is equivalent to a saturated proportional control law, effectively mapping $e \in R$ into $x_r(e) \in [-\pi/2, \pi/2]$. As shown in the geometry of Figure 5.20, a small lookahead distance implies aggressive steering, which intuitively is confirmed by a correspondingly large proportional gain in the saturated control interpretation. However because a sailboat is an underactuated vessel that can only steer by attitude information and is subject to influence like ocean currents and nonzero slip angles B . This suggest that a integral controller will be a needed for the sailboat to follow straight-line. The integral controller is shown below,

$$x_r(e) = \arctan \left(-K_p e - K_i \int_0^t e(\tau) d\tau \right) \quad (5.85)$$

with $K_i > 0$. Considering horizontal path following along straight lines, the desired yaw angle can be computed by,

$$x_d(e) = \alpha_k + x_r(e) \quad (5.86)$$

with $x_r(e)$ as in Equation 5.84. In practice, to avoid overshoot and windup affects, care must be taken when using integral action in the steering law. Specifically, the integral term should only be used when a steady-state off-track condition is detected.

5.4.4. Path-Following Controllers

The path-following controller depends on having access to velocity measurements. This methods aim is to align the velocity and LOS vector. The desired course angle x_d is computed such that the velocity vector is along the path(LOS vector) using the *lookahead-based steering* law,

$$x_d(e) = x_p + x_r(e) = \alpha_k + \arctan(-K_p e) \quad (5.87)$$

The control objective $x \rightarrow x_d$ is satisfied by transforming the course angle command x_d to a heading angle command ψ_d . This requires the knowledge of the β since, illustrated in Figure 5.21,

$$\psi_d = x_d - \beta \quad (5.88)$$

The velocity and LOS vectors can be aligned using a heading controller with the following error signal,

$$\psi = \psi - \psi_d = \psi - x_d + \beta \quad (5.89)$$

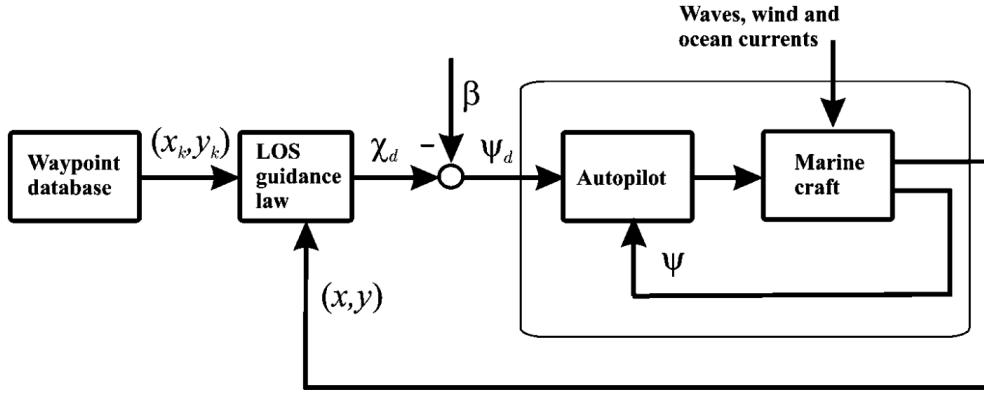


Figure 5.21: LOS guidance principle where the side slip angle B can be applied and compensated for by using integral action

If the velocities of the vessel are measured, the sideslip angle can be computed by

$$\beta = \arcsin\left(\frac{v}{U}\right) \quad (5.90)$$

Guidance laws of PI type avoid velocity measurements by treating β as an unknown slowly varying disturbance satisfying $\beta \approx 0$.

To ensure that the reference angle ψ_d , that is fed into the autopilot, is always the shortest angle between the waypoint and vessel position, the waypoint and vessel position are transformed into vectors and the shortest angle between the vectors are calculate. To find the angle between two vectors the following equation is used, which is derived using the Pythagorean Theorem,

$$\theta = \cos^{-1}\left(\frac{\bar{u} \cdot \bar{v}}{\|\bar{u}\| \|\bar{v}\|}\right) \quad (5.91)$$

The vectors \bar{u} and \bar{v} is illustrated in Figure 5.22. The vector \bar{u} is defined using the reference angle ψ_d and the vector \bar{v} is defined using the heading angle ψ .

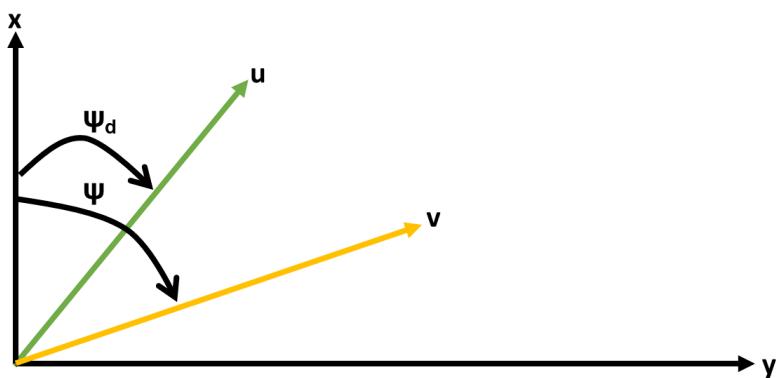


Figure 5.22: Definition of vector \bar{u} and \bar{v}

Vector \bar{u} and \bar{v} is decomposed into its x- and y-components by the following equation,

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos(\psi) \\ \sin(\psi) \end{bmatrix} \quad (5.92)$$

The vector components are used to calculate the dot product and the magnitude for each vector as shown below,

$$\bar{u} \cdot \bar{v} = x_{\bar{u}} x_{\bar{v}} + y_{\bar{u}} y_{\bar{v}} \quad (5.93)$$

$$\|\bar{u}\| = \sqrt{x_{\bar{u}}^2 + y_{\bar{u}}^2} \quad (5.94)$$

$$\|\bar{v}\| = \sqrt{x_{\bar{v}}^2 + y_{\bar{v}}^2} \quad (5.95)$$

The calculated angle θ , which is the shortest angle between the two vectors, is now fed into the autopilot with the following transformation

$$\psi_d = -\theta \quad (5.96)$$

5.4.5. Circle of Acceptance for Surface Vessel

When a vessel is traveling along a straight line piece wise path, made up of n straight-line segments connected by $n + 1$ waypoints. A mechanism is required for selecting the next waypoint. The next waypoint (x_{k+1}, y_{k+1}) can be selected on the basis of whether or not the vessel lies within a *circle of acceptance* with radius R_{k+1} around (x_{k+1}, y_{k+1}) . This means that the vessel position must at time t satisfy,

$$[x_{k+1} - x(t)]^2 + [y_{k+1} - y(t)]^2 \leq R_{k+1}^2 \quad (5.97)$$

then the next waypoint (x_{k+1}, y_{k+1}) should be selected. A suggested method is to select the radius R depending on the distance between the two waypoints.

$$R = b \frac{y_{k+1} - y_k}{x_{k+1} - x_k} \quad (5.98)$$

where the value b is a constant value.

5.5. Sailboat Maneuvering

5.5.1. The Three forms of Sailboat Maneuvering

The maneuvering of a sailboat comes down to three different types of maneuvering [32]. The maneuverings are called tacking, jibbing and normal sailing. The tacking maneuver is used when the sailboat is required to sail into the wind, where the sailboat switches between sailing close hauled left and right, which is illustrated in Figure 5.23. The tacking maneuver is illustrated in Figure 6.8.

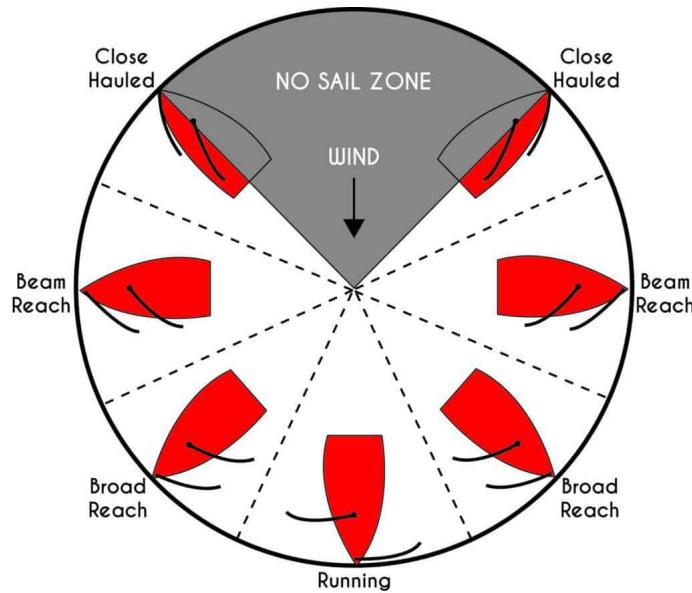


Figure 5.23: Sail Zone

The jibbing maneuver is when the sailboat sails downwind and is known as running, illustrated in Figure 5.23. The sail angle will switch between 90° and -90° , with a small rudder angle change. The act of switching the sail angle is what is known as jibbing. The last maneuver, which is normal sailing, is merely the control of the rudder and sail according to the direction of the wind and the direction in which it desires to sail in. The decision tree for the three maneuvers can be presented in the form of a finite state machine.

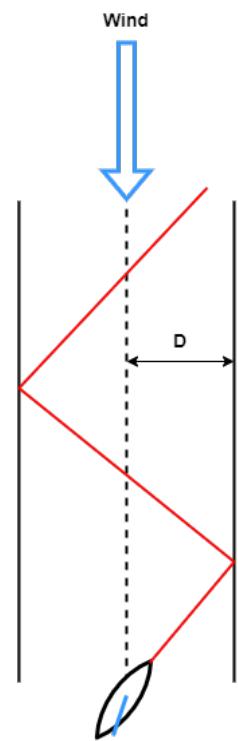


Figure 5.24: Tacking

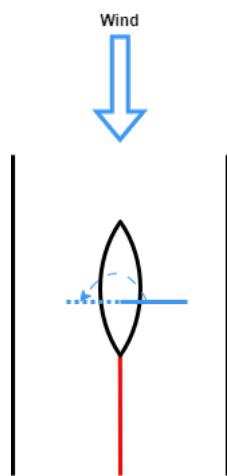


Figure 5.25: Jibbing

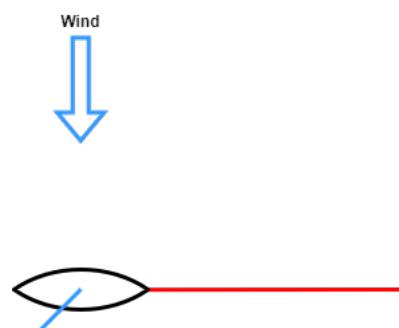


Figure 5.26: Normal Sailing

5.5.2. Finite State Machine of Sailboat Maneuvering

The three maneuver abilities of a sailboat can be implemented via a state machine, more specifically a Moore state machine. Where the states are only dependent on the input actions. The input action in this case are the apparent wind angle, defined in Equation 4.37.

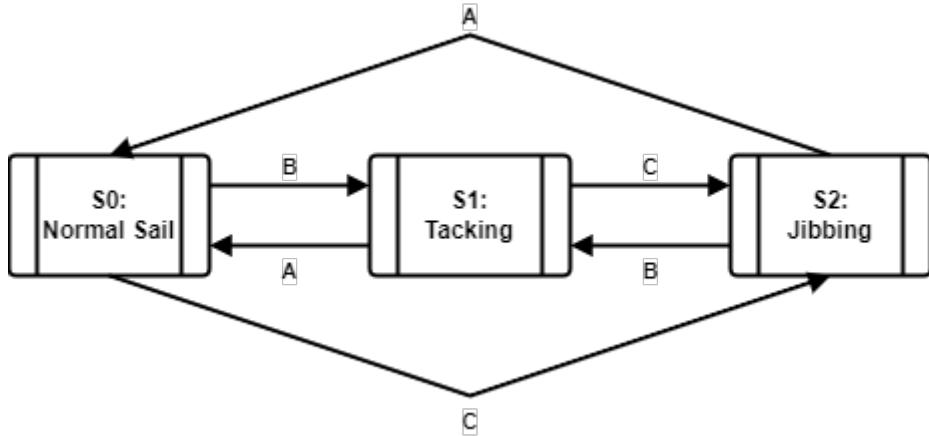


Figure 5.27: Finite-State Machine for Sailboat Maneuvering

The state machine is illustrated in Figure 5.27. The actions are stated as **A**, **B** and **C** and is defined in Table 5.1. The angle are only expressed in the case of a positive heading angle.

Table 5.1: Definitions of Actions in State Machine 5.27

Action	Definition
A	$45 < \text{Apparent Wind Angle} < 315$ $!(150 < \text{Apparent Wind Angle} < 210)$
B	$\text{Apparent Wind Angle} < 45$ $\text{Apparent Wind Angle} > 315$
C	$150 < \text{Apparent Wind Angle} < 210$

5.5.3. Heading Controller

The heading controller, discussed in Section 5.5.2, is responsible for determining which state of sailing the boat is in and also adjusting the the tacking angle with an offset of -45° and 45° from the apparent wind angle. The pseudo code of the controller is shown in Algorithm 5.1 and the implementation in Matlab is illustrated in the Appendix Subsection A.5.2.

For state 1, tacking, another finite-state machine will define the inner workings that dictates the actions of the desired heading angle. The heading angle will then be sent to the low level control which dictates the rudder angle required to reached the desired

Algorithm 5.1: Heading Controller

```

 $\gamma$  = Wind angle
if  $\gamma > 180$  then
     $\gamma = \gamma + 360$ 
else if  $\gamma < -180$  then
     $\gamma = \gamma - 360$ 
end if
if (absolute value of  $\gamma$ ) < 45 then                                ▷ State 1
    if tack heading < 15 then
        State = Initialized right tack
        Set tack distance
        Set tack right and tack left angle
        Reference Heading = tack right
    else
        State = Tacking
        Set tack distance
        Set tack right and tack left angle
        Reference Heading = Tack left/Tack Right
    end if
else                                                               ▷ State 0
    State = Normal Sailing
    Reference Heading = Waypoint Heading
end if

```

heading. For states 0, normal sail, and 2, jibbing, the heading angle is merely passed onto the rudder controller without the need for another state machine.

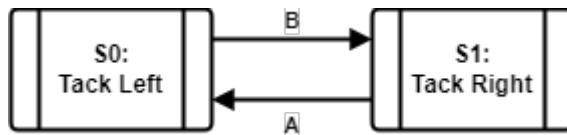


Figure 5.28: Finite-State Machine for Sailboat Maneuvering

The actions definitions for the tacking state machine is illustrated in Figure 5.28. The e is the y -component of the guidance axis, defined in Section 5.4.1, and the Y is the value that corresponds the the width, D , illustrated in Figure 6.8. The optimal heading angle for tack left is -45° and tack right is 45° .

Table 5.2: Definitions of Actions in State Machine 5.28

Action	Definition
A	$e > D$
B	$e < -D$

The sailboat actions during a tack is summarized in below,

1. Tack Right

- (a) Set reference heading to 45°
 - (b) While the sailboat turns winch in the sail accordingly
 - (c) Once the sailboat is sailing in the correct direction release the sail to the correct angle
2. Tack Left
- (a) Set reference heading to -45°
 - (b) While the sailboat turns winch in the sail accordingly
 - (c) Once the sailboat is sailing in the correct direction release the sail to the correct angle

There will be two rudder controllers, one controller will be used to adjust the rudder angle depending on the reference heading and then a second controller will be used only in the state of tacking. The controller is fuzzy logic controller that switches the reference heading depending on the cross-track error.

5.5.4. Tacking Controller

Direct Tack

The tacking controller for a direct tack maneuver is a fuzzy logic controller that has the inputs tack direction, tack distance, tack left angle and tack right angle. The pseudo code of the fuzzy logic controller is shown below,

Algorithm 5.2: Fuzzy Logic Tacking Controller

Require: State 1: Tacking

```

 $\gamma$  = Apparent wind angle
if  $e > D$  then
   $\psi_d = 45 + \gamma$                                  $\triangleright$  Tack Right Angle
else if  $e < -D$  then
   $\psi_d = -45 + \gamma$                              $\triangleright$  Tack Left Angle
end if

```

The value of D can be adjusted to change the amount of tacks being performed to reach a waypoint. The adjustment happens inside the heading controller, discussed in Section A.5.2.

Indirect Tack

A indirect tack maneuver is where the sailboat only tacks twice to reach a waypoint when sailing into the wind. Indirect tacking will favour speed due to the sailboat gaining speed the longer it stays on a tack and losing speed when a tack is performed.

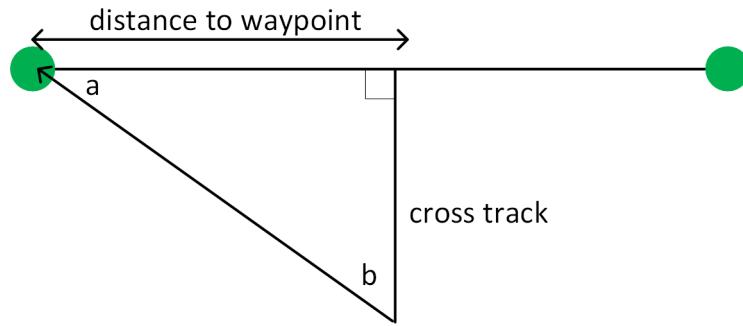


Figure 5.29: Indirect tack calculations

The sailboat will tack back to the destination waypoint when either one of the two angles a/b is 45° . The equation below illustrates how the angle can be calculated,

$$b = \tan^{-1} \left(\frac{\text{cross track}}{\text{distance to waypoint}} \right) \quad (5.99)$$

Chapter 6

Ardupilot

This chapter gives a overview of the Ardupilot software, more specifically the rover firmware. The sailboat is a extension of the rover firmware, although there is a sailboat option there is still limited functionality. This chapter will also introduce the Ardupilot code flow and then give detailed explanation on extensions made to improve or add functionality.

6.1. Architecture

Before looking at the architecture of Ardupilot you need to understand exactly what Ardupilot is [33]. Ardupilot is what is known as an autopilot, a system that automatically controls a vehicle. The control can consists of speed control, waypoint control and altitude control to name a few. Ardupilot is created for the use in multiple vehicles with a list below,

- Plane
- Copter
- Rover
- Sub
- Blimp
- AntennaTracker

The above vehicles are known as vehicle types and each type further has specific frameworks that is specific to a type of plane, copter or rover. Each framework has different actuators and control abilities. Inside the rover vehicle type the framework of interest, sailboat, resides. Although a sailboat is not a type of rover due to the its novelty it does not have its own vehicle type but rather makes use of the rover class.

The high level architecture of the ArduPilot software is illustrated below. As seen in Figure 6.1 the Ardupilot is responsible for three things, the vehicle specific code, the sensor libraries and the hardware abstraction layer(HAL).

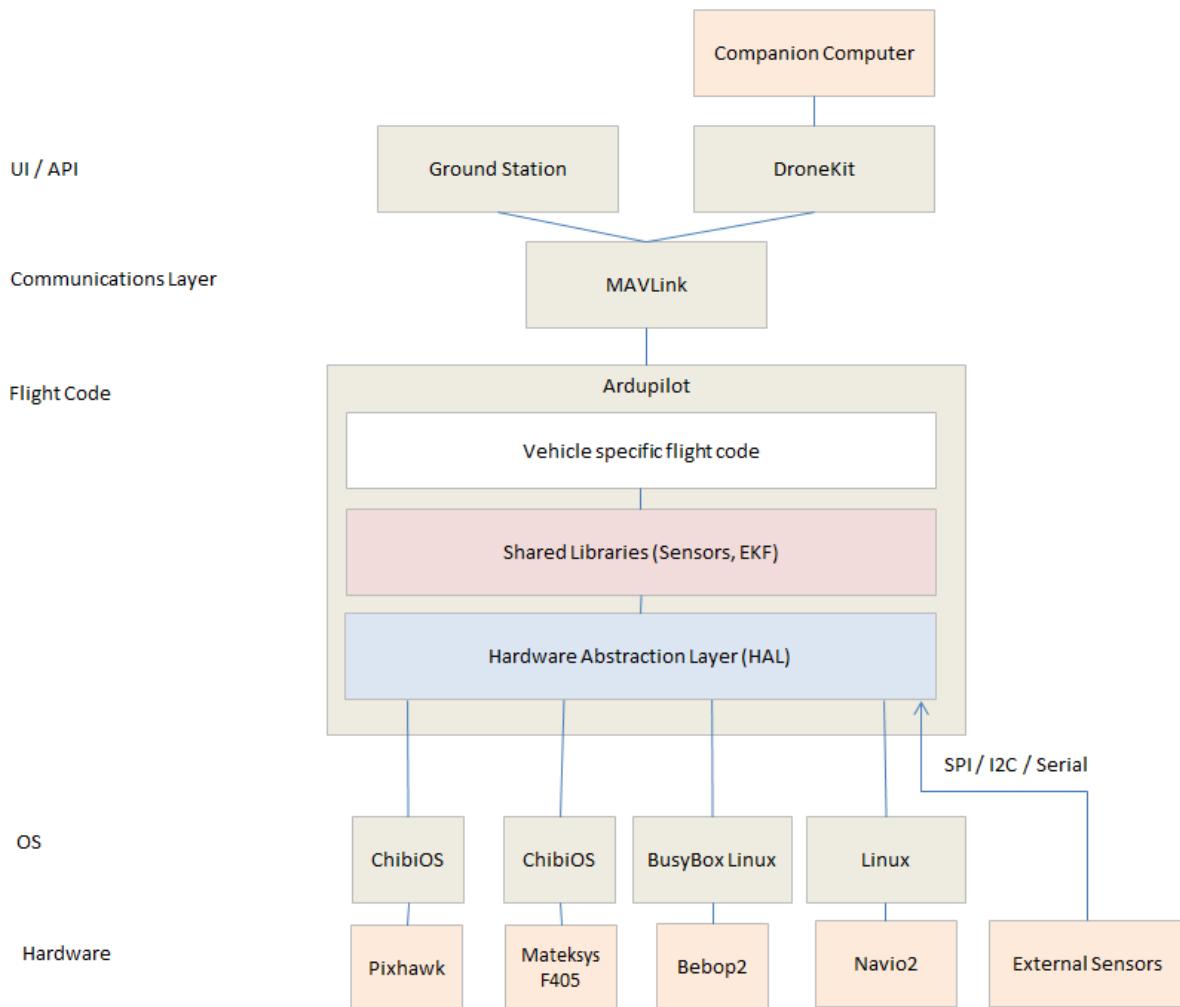


Figure 6.1: High Level Architecture of Ardupilot Software

6.2. Rover Architecture

Now that the high level overview of where Ardupilot fits in is explained a deep dive into specifically the rover code is required. The rover code architecture is illustrated in Figure 6.2. The rover architecture shows the background threads are running the sensors defined in the libraries class. The main loop is where the rover code is initialized and also where the scheduler runs, which will be discussed in detail later in this chapter. The Mode class is important and it is inside the mode that the specific framework of the sailboat will be called from. The mode determines the how the sailboats rudder and sail will be controlled

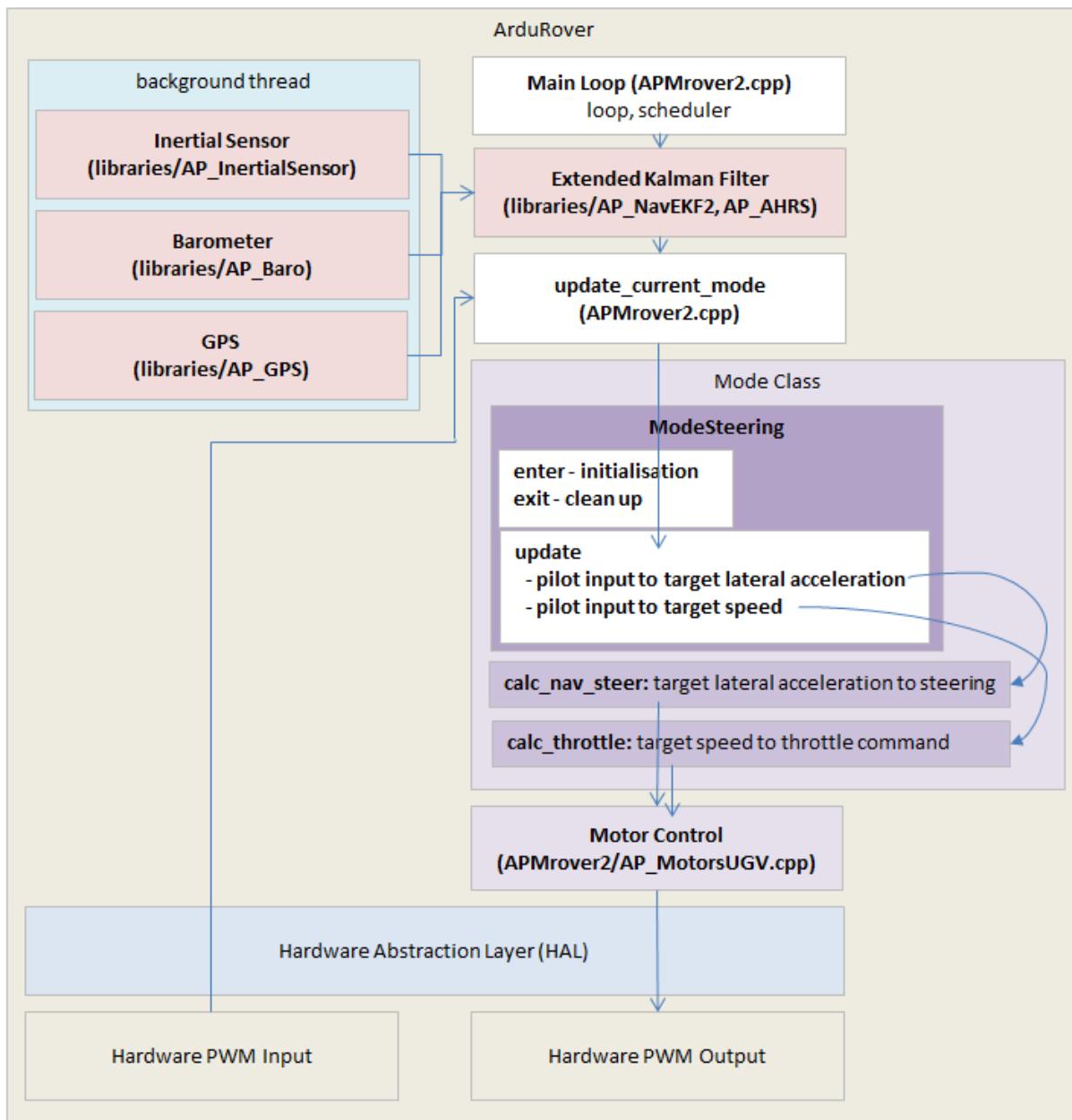
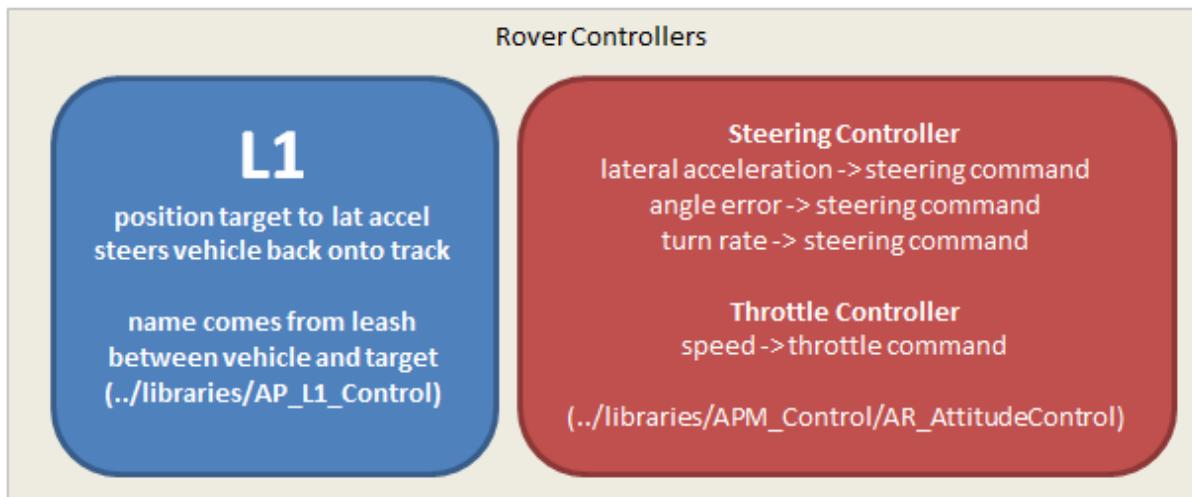


Figure 6.2: Rover architecture

The rover vehicle has three high level controllers, L1 controller, Steering Controller and Throttle controller. The three controllers are illustrated in Figure 6.3. The L1 controller converts an origin point and a destination point into a lateral acceleration to make the vehicle travel along the path from the origin to the destination. This lateral acceleration is then passed to the steering controller.

**Figure 6.3:** Rover Controllers

The steering controller converts the desired lateral acceleration, the angle error or the desired turn rate into a steering output command that is fed into the motor library. The throttle controller converts the desired speed into a throttle command that is fed into the motor library. For the sailboat a new controller was designed that will calculate the optimal sail angle.

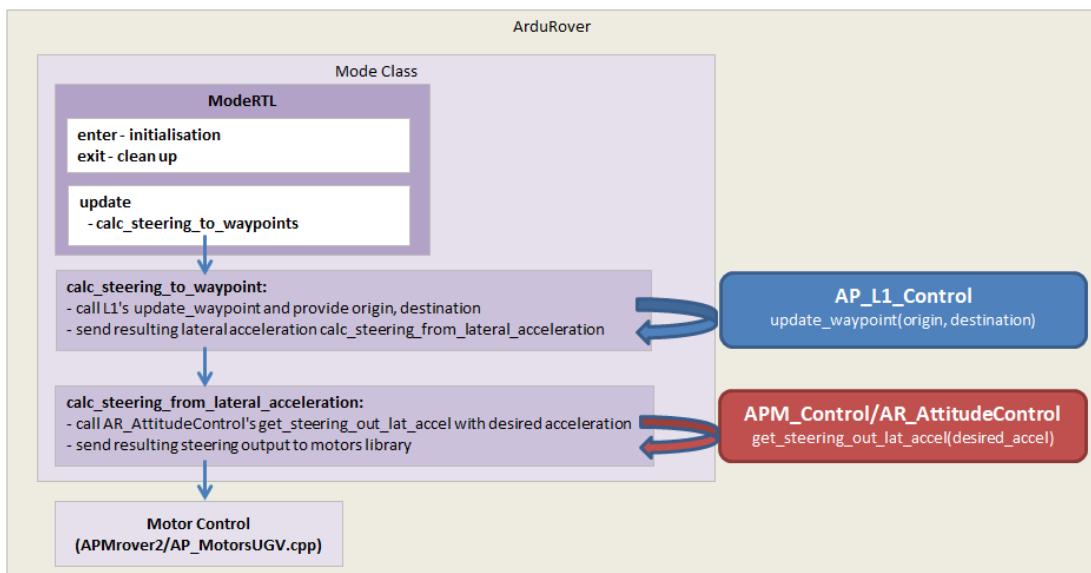
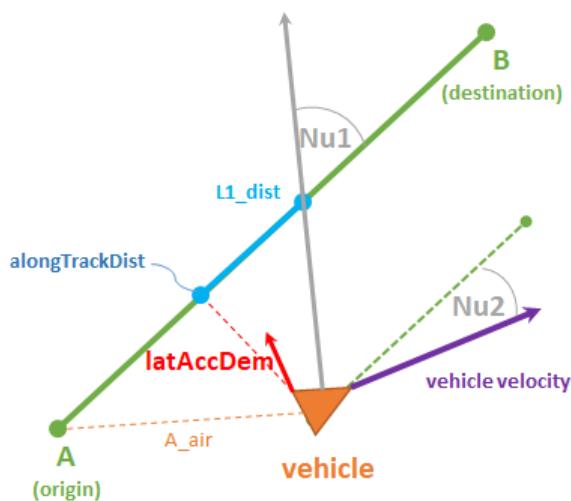
**Figure 6.4:** Rover navigation overview

Figure 6.4 gives an overview of how the rover navigation architecture works. On every iteration of the main loop (50hz) a call is made to the active mode's update method. While in Auto, Guide, RTL and SmartRTL mode, the update calls into the Mode class's **calc_steering_to_waypoint** method. Mode's **alc_steering_to_waypoint** then calls the **AP_L1_controller** library's **update_waypoint** method providing it the location that the rover should drive towards. The **AP_L1_controller**'s **update_waypoint** method

returns a desired **lateral_acceleration** which is passed into Mode's **calc_steering_from_lateral_acceleration**. Mode's **calc_steering_from_lateral_acceleration** sends the desired acceleration to **APM_Control/AR_AttitudeControl**'s **get_steering_out_lat_accel** which uses a PID controller to calculate a steering output. The steering output is sent into the **AP_MotorsUGV** library using the **set_steering** method. The final output of the L1 controller's **update_waypoint** method is a desired lateral acceleration which should bring the vehicle back to the line between the origin and destination. The formulas used are shown below,



alongTrackDist: distance from A to closest point on line from A to B

L1_dist: distance from alongTractDist to target point on path

$$L1_dist = 0.3 * damping * period * speed$$

Nu1: angle from vehicle to L1_dist point (relative to line from A to B)

Nu2: vehicle velocity angle relative (relative to line from A to B)

latAccDem: desired acceleration output along red cross track line

$$LatAccDem = \frac{4 * damping^2 * speed^2 * \sin(Nu1 + Nu2)}{L1_dist}$$

Figure 6.5: Formulas for L1 controllers

6.3. Sailboat Framework

The sailboat framework inside the rover vehicle type has the following modes to choose from:

- Manual
 - The sail and rudder is directly controller via the transmitter input. The minimum throttle value sets the sail to fully in and the maximum throttle releases the sail fully. This mode is used as a fail safe when an error or problem occurs in the other modes that controls the sailboat. A GPS position cannot be given to guide the sailboat.
- Acro
 - In Acro mode the user's steering stick controls the vehicle's turn rate and the throttle stick controls the vehicle's speed. The sail is automatically trimmed to the wind direction using to the wind vane. The sailboat will only try to sail if the transmitter gives a forward speed. With a 0 throttle input the sailboat will let out the sail to attempt to stop. This mode will only be used to estimate parameters of the sailboat that will be used to configure the steering control.
- Loiter
 - Loiter mode sets the sailboat to stay in the same position. If the sailboat does not have any speed and is within the Loiter radius, the sailboat simply drifts otherwise the sailboat will perform a figure 8 type of motion to keep its position.
- RTL
 - RTL stands for Return-To-Launch. The sailboat will return to the home position that is set on the ground control software. The sailboat will tack if it is required to reach the home position. Once it reaches the home position it will loiter around the home position.
- Auto
 - In Auto mode the sailboat will follow a pre-programmed mission that is stored in the ground control software. The mission consists of waypoints which the sailboat will follow in its numerical order and the sailboat will tack if required to reach a waypoint.
- Guided

- In guided mode the sailboat is controlled via the ground control software by giving the sailboats points to sail towards. The sailboat can tack if required.

The mode that will be used is the auto mode. The auto mode's code execution flow is illustrated in Figure 6.6. The *Rover.cpp* file is the main code where the system is initialized and the scheduler(task manager) is executed from. The scheduler executes the rover's functionalities, such as the reading sensor data, sending commands to the actuators, changing modes and storing the telemetry data.

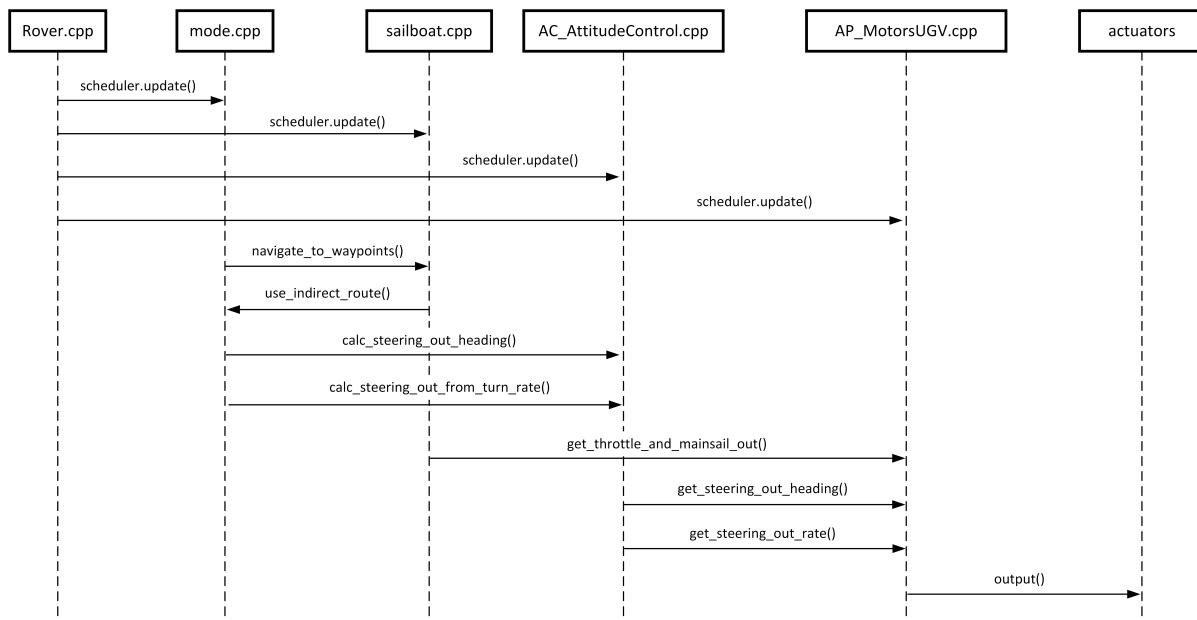


Figure 6.6: Code Execution Flow for Sailboat in auto mode

The *mode.cpp* is responsible for the different modes of operating listed above. The *navigate_to_waypoint()* method manages the mission points. The method acts a sort of state machine feeding the next waypoint as soon as the current waypoint is reached, which is updated through the *update()* method. The sailboats orientation is controlled through the methods *calc_steering_out_heading()*. The heading is calculated by drawing a straight line between two waypoints, or a waypoint and the source. The steering method then calls *AC_AttitudeControl* class, which returns the required positions of the actuators and then the value is passed to the methods of the *AP_MotorsUGV* class that converts the control signals into the appropriate values for the actuators.

6.3.1. The Shortcomings of Sailboat Framework

- Sail Controller
- Roll Controller through rudder/sail
- Rudder angle, sail angle missing from log data

6.4. Ardupilots Control Strategies

6.4.1. Sail Controller

Ardupilots implementation of a sail controller is still in its infancy. The controller merely adjusts the sail with an offset to the apparent wind angle. The sail controller requires a wind directional sensor to work so there is no back-up sail controller if the wind sensor malfunctions or is unavailable.

Algorithm 6.3: Ardupilots Sail Controller

γ_ω = Apparent Wind angle

y = Sail Angle

u = Ideal Sail Angle

$y = \gamma_\omega - u$

The controller can be expanded to dynamically adjusting the sail to achieve maximum speed in all wind directions or a extreme seeking controller, discussed in [28], where the sail controller controls the sail angle independently of the wind angle by seeking the optimal sail angle. Also what can be very beneficial is a VPP that allows the user to control the speed of the sailboat through the sail angle.

6.4.2. Rudder Controller

The rudder control has four options a P controller, PI controller, PD controller and PID controller. The controller has a negative feedback of the actual heading angle and the input the reference heading angle minus the actual heading angle. Then there is also an option to set a fixed turn rate on the rudder.

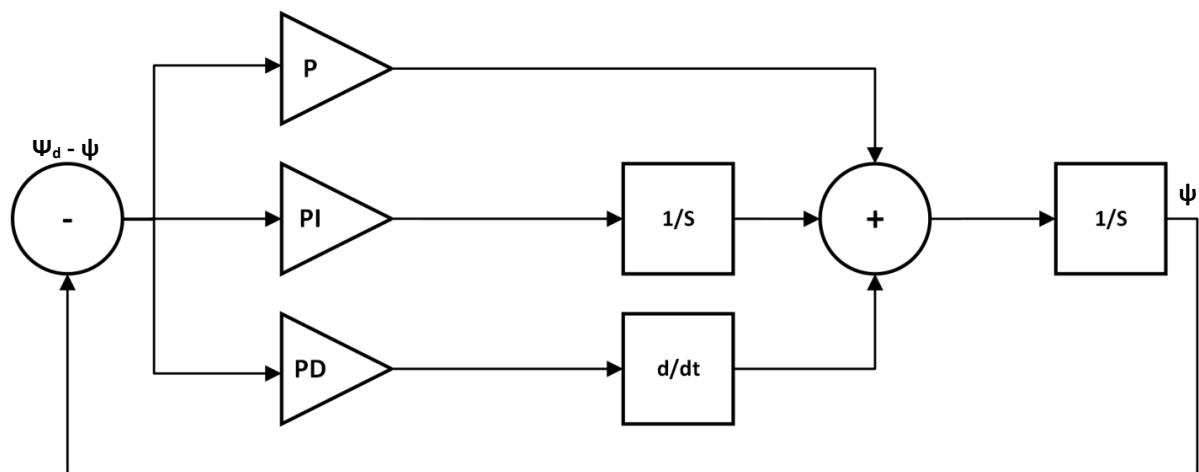


Figure 6.7: Ardupilots Rudder Controller

Expanding the rudder control to non-linear controller that has variable gain depending on

the error. Or a different rudder controller during tacking to ensure a faster transition between tacks.

6.4.3. Roll Controller

At the moment of conducting my research there is no roll controller available in Ardupilot. There is some obvious ways to incorporate a roll controller either by using the rudder or the sail.

6.4.4. Tacking

Ardupilots version of tacking is pretty straightforward, where a corridor is defined from the previous waypoint to the next waypoint, and if that waypoint corridor is in the no go zone, the sailboat will tack. The width of the corridor is the only changeable variable when tacking. The Figure below illustrates a tack straight into the wind.

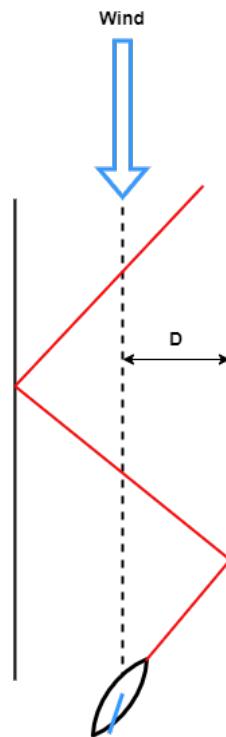


Figure 6.8: Tacking

Improvements on the taking can be to switch between fast and precise tack, where fast tack is when the sailboat tacks as little as possible when sailing update and precise tacking is when the sailboat tacks as much as possible to keep the corridor small. Might be a good idea to create a new sailing mode.

6.4.5. Waypoint Navigation

Ardupilots waypoint navigation works by defining waypoints in mission planner and then writing it to the controller. The controller works like a state machine switching to the next waypoints once the current waypoint is reached. You can defined radius R, illustrated in Figure 6.9, as a radius of acceptance for when a waypoint is reached. The L1 controller is responsible for steering the boat the line connecting the two waypoints, previous and current waypoint that is.

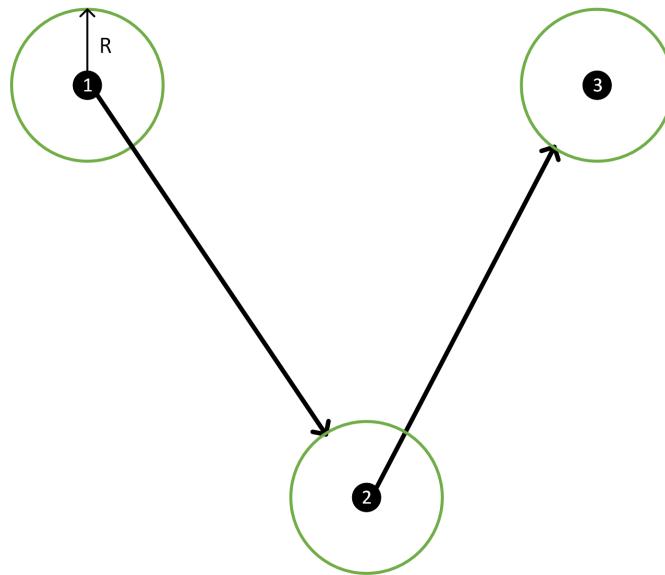


Figure 6.9: Ardupilots Waypoint Navigation

Chapter 7

Simulations

In this chapter the simulations and their results are discussed. The first simulation is done in Matlab with an accurate non-linear model of a sailing yacht simulated in 4-DOF. Onto this model of the vessel a guidance system, sailboat maneuvering, waypoint navigation, rudder, sail, roll the control strategies and was implemented. The second simulations are used to test the firmware, running in the autopilot, in a SITL and HITL environment.

7.1. RSS Simulator

The simulation, as mentioned, is based on the work done in [13]. Firstly before modifying the simulator the simulator needs to be thoroughly inspected and described. The rigid-body dynamics used in the simulator is Fossen's compact vectorial expression shown below,

$$\mathbf{M}_{RB}\dot{\mathbf{v}} + \mathbf{C}_{RB}(\mathbf{v})\mathbf{v} = \boldsymbol{\tau}_{RB} \quad (7.1)$$

The equation is explained in Chapter ???. The simulation used a simplified version where the matrix M_A , due to a sail yacht's symmetry about the x_bz_b -plane and $M_{Aij} = M_{Aji}$, is defined as,

$$\mathbf{M}_A = \begin{bmatrix} X_{\dot{u}} & 0 & 0 & 0 \\ 0 & Y_{\dot{v}} & 0 & 0 \\ 0 & 0 & K_{\dot{p}} & 0 \\ 0 & 0 & 0 & N_{\dot{r}} \end{bmatrix} \quad (7.2)$$

The $C_A(v)$ matrix is also simplified to,

$$C_A(v) = \begin{bmatrix} 0 & 0 & 0 & -Y_{\dot{v}}v \\ 0 & 0 & 0 & Y_{\dot{u}}u \\ 0 & 0 & 0 & 0 \\ Y_{\dot{v}}v & Y_{\dot{u}}u & 0 & 0 \end{bmatrix} \quad (7.3)$$

Lastly the M_{RB} matrix is simplified to $M_{RB} = \text{diag}\{m, m, I_{xx}, I_{zz}\}$.

Now for the addition made to the simulation, RSS (Ruben's Sailboat Simulator). The simulator came equipped with a very basic linear and non-linear heading controller. The controller made used of the back stepping method, explained in [34]. The simulator had no other form of control.

7.1.1. Additions in RSS

Firstly a more complete sail controller and a PID heading controller. These low level controllers are responsible for controller the rudder and sail. The high level control implemented consists of a tacking controller, indirect method to tacking, a controller controller when the sailboat should tack and when not, waypoint scheduler and navigation system. A high level block diagram overview showing the inflow of information only is illustrated in Figure 7.1. The green blocks indicate the additions made to the simulator.

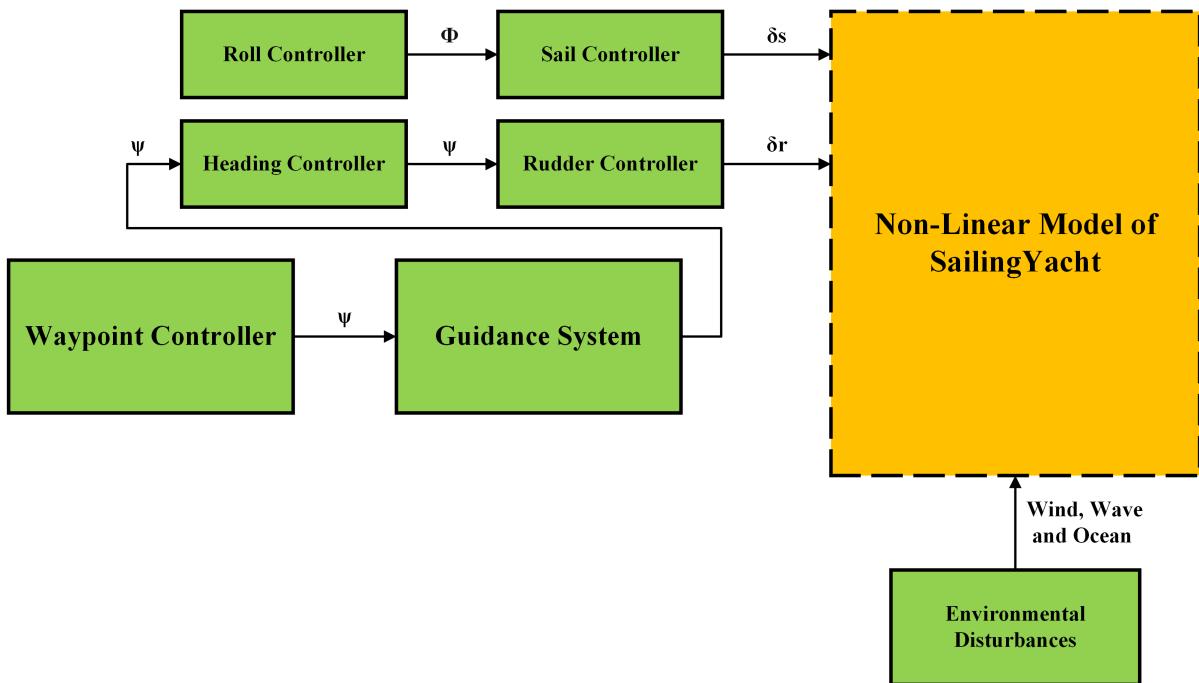


Figure 7.1: High level block diagram overview

Chapter 8

Results

Chapter 9

Conclusion

Bibliography

- [1] R. van Tonder, “Github repository,” <https://github.com/22569596>, 2023.
- [2] G. Kilpin, “Modelling and design of an autonomous sailboat for ocean observation,” Master’s thesis, University of Cape Town, Rondebosch, Cape Town, 7700, Sep. 2014.
- [3] M. Tranzatto, “Navigation and control for an autonomous sailing model boat,” Master’s thesis, University of Pisa, Lungarno Pacinotti 43, Italy, Sep. 2016.
- [4] K. L. Wille, “Autonomous sailboats,” Master’s thesis, Norwegian University of Science and Technology, Trondheim Ålesund Gjøvik, Norway, Jun. 2016.
- [5] Y.-W. C. J. W. S.-Y. C. Yee-Teng Ang, Wey-Kean Ng and L. B. Firth, “An autonomous sailboat for environment monitoring,” *Thirteenth International Conference on Ubiquitous and Future Networks*, pp. 242–246, 2022.
- [6] M. A.-C. W. S. R. D. W. Joga Dharma Setiawan, Deddy Chrismianto and S. Alimi, “Development of dynamic model of autonomous sailboat for simulation and control,” *International Conference on Information Technology, Computer and Electrical Engineering*, no. 7, pp. 52–57, 2020.
- [7] Saildrone, “About us,” <https://www.saildrone.com/about>, 2023.
- [8] H. Model, “Joysway dragon flite 95 v2 racing sailing yacht artr,” <https://howesmodels.co.uk/product/joysway-dragon-flite-95-v2-racing-sailing-yacht-artr/>, 2023, sailboat RC model.
- [9] M. Oborne, “Mission planner,” <https://ardupilot.org/planner/>, 2023.
- [10] Holybro, “Holybro pixhawk 6c,” http://docs.px4.io/main/en/flight_controller/pixhawk6c.html, 2023, pixhawk 6C.
- [11] ——, “Holybro m8n gps,” <https://holybro.com/collections/standard-gps-module>, 2023, m8N GPS.
- [12] Dronecode, “Mavlink protocol,” <https://mavlink.io/en/>, 2023, mAVLink.
- [13] L. Xiao and J. Jouffroy, “Modeling and nonlinear heading control for sailing yachts,” *IEEE Journal of Oceanic Engineering*, vol. 39, no. 2, pp. 256–268, 2014.

- [14] T. I. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control*. John Wiley Sons Ltd, 2011.
- [15] K. D. Do and J. Pan, *Control of Ships and Underwater Vehicles*. Springer Science+Business Media, LLC, 2009.
- [16] R. Featherstone, *Rigid Body Dynamics Algorithms*. Springer Science+Business Media, LLC, 2008.
- [17] O. M. Faltinsen, *Sea Loads on Ships and Offshore Structures*. Cambridge University Press, 1990.
- [18] W. Blödnermann, “Parameter identification of wind loads on ships,” *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 51, no. 3, pp. 339–351, 1994.
- [19] K. H. Son and K. Nomoto, “On the coupled motion of steering and rolling of high-speed container ship,” *Naval Architecture and Ocean Engineering*, vol. 20, pp. 73–83, August 1982.
- [20] A. F. Molland and S. R. Turnock, *Marine Rudder and Control Surface*. Butterworth-Heinemann, 2021.
- [21] K. Legursky, “System identification and the modeling of sailing yachts,” Ph.D. dissertation, University of Kansas, 2013.
- [22] D. Morris, “Derivation of forces on a sail using pressure and shape measurements at full-scale,” Master’s thesis, CHALMERS UNIVERSITY OF TECHNOLOGY, Chalmersplatsen 4, 412 96 Göteborg, Sweden, Sep. 2011.
- [23] L. Larsson and R. E. Eliasson, *Principles of Yacht Design*. International Marine/McGraw-Hill, 2007.
- [24] A. Gentry, “The aerodynamics of sail interaction,” *AIAA Symposium on the Aero/Hydronautics of Sailing*, vol. 1, no. 1, p. 8, 1971.
- [25] Y. Masuyama and T. Fukasawa, “Tacking simulation of sailing yachts with new model of aerodynamic force variation during tacking maneuver,” *Journal of Sailboat Technology*, vol. 1, no. 1, pp. 1–34, 2011.
- [26] L. Zhou, K. Chen, Z. Chen, H. Dong, and D. Song, “Course control of unmanned sailboat based on bas-pid algorithm,” *International Conference on System Science and Engineering*, August 2020.
- [27] S. N. H. Abrougui and H. Dallagi, “Modeling and autopilot design for an autonomous catamaran sailboat based on feedback linearization,” *International Conference on Advanced Systems and Emergent Technologies*, vol. 1, no. 1, pp. 130–135, 2019.

- [28] D. H. dos Santos, “Framework for comparing autonomous sailboat control systems,” Ph.D. dissertation, Federak University of Rio Grando do Norte, 202.
- [29] J. A. A. Engelbrecht, “Advance automation 833: Introductory course to aircraft dynamics,” April 2019, aircrafts.
- [30] L. E. Dubins, “On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents,” *American Journal of Mathematics*, vol. 79, no. 3, pp. 497–516, 1957.
- [31] M. Breivik and T. I. Fossen, “Path following for marine surface vessels,” *Centre for Ships and Ocean Structures (CESOS*, no. 1, pp. 2282–2289, 2004.
- [32] C. C. German, “Essential sailing maneuvers how to master them,” <https://www.lifeofsailing.com/post/essential-sailing-maneuvers>, 2023, sail Maneuvering.
- [33] ArduPilot, “Ardupilot development,” <https://ardupilot.org/dev/index.html>, 2023, ardupilot.
- [34] P. Kokotovic, “The joy of feedback: nonlinear and adaptive,” *IEEE Control Systems Magazine*, vol. 12, no. 3, pp. 7–17, June 1992.

Appendix A

Additional Modelling Information

A.1. Notation and Vector Definitions

Table A.1: Rigid body motion vectors

Vector	Components	Definition
\mathbf{f}_{Ob}	$[X \ Y \ Z]^T$	force decomposed in the body-fixed frame
\mathbf{m}_{Ob}	$[K \ M \ N]^T$	moment decomposed in the body-fixed frame
\mathbf{v}_{Ob}	$[u \ v \ w]^T$	linear velocity decomposed in the body-fixed frame
\mathbf{w}_{Ob}^E	$[p \ q \ r]^T$	angular velocity of the body-fixed relative to the earth-fixed frame
\mathbf{r}_{Ob}	$[x_g \ y_g \ z_g]^T$	vector from O_b to CG decomposed in the body-fixed frame

A.2. Modeling Equations

A.2.1. Rigid-Body System Inertia Matrix \mathbf{M}_{RB}

The rigid-body system inertia matrix \mathbf{M}_{RB} is unique and satisfies the following properties,

$$\mathbf{M}_{RB} = \mathbf{M}_{RB}^T > 0, \quad \dot{\mathbf{M}}_{RB} = \mathbf{0}_{6 \times 6} \quad (\text{A.1})$$

where

$$\begin{aligned} \mathbf{M}_{RB} &= \begin{bmatrix} m\mathbf{I}_{3 \times 3} & -m\mathbf{S}(\mathbf{r}_g^b) \\ m\mathbf{S}(\mathbf{r}_g^b) & \mathbf{I}_b \end{bmatrix} \\ &= \begin{bmatrix} m & 0 & 0 & mz_g & mz_g & -my_g \\ 0 & m & 0 & 0 & 0 & mx_g \\ 0 & 0 & m & -mx_g & -mx_g & 0 \\ 0 & -mz_g & -my_g & I_x & -I_{xy} & -I_{xz} \\ mz_g & 0 & -mx_g & -I_{xy} & I_y & -I_{yz} \\ -my_g & mx_g & 0 & -I_{zx} & -I_{zy} & I_z \end{bmatrix} \end{aligned} \quad (\text{A.2})$$

Here $\mathbf{I}_{3 \times 3}$ is the identity matrix, $\mathbf{I}_b = \mathbf{I}_b^T > 0$ is the inertia matrix and $\mathbf{S}(\mathbf{r}_g^b)$ is a skew-symmetric matrix.

A.2.2. Rigid-Body Coriolis and Centripetal Matrix \mathbf{C}_{RB}

The rigid-body Coriolis and centripetal matrix $\mathbf{C}_{RB}(\mathbf{v})$ can always be represented such that $\mathbf{C}_{RB}(\mathbf{v})$ is skew-symmetric. Moreover it satisfy the following property,

$$\mathbf{C}_{RB}(\mathbf{v}) = -\mathbf{C}_{RB}^T(\mathbf{v}). \quad \forall \mathbf{v} \in \mathbf{R}^6 \quad (\text{A.3})$$

The skew-symmetric property is very useful in the design of nonlinear motion control systems since the quadratic form $\mathbf{v}^T \mathbf{C}_{RB}(\mathbf{v}) \mathbf{v} \equiv 0$. The rigid-body Coriolis and centripetal terms can be expressed in component form shown below,

$$\mathbf{C}_{RB}(\mathbf{v}) = \begin{bmatrix} 0 & 0 & 0 & m(y_g q + z_g r) & -m(x_g q - w) & -m(x_g r + v) \\ 0 & 0 & 0 & -m(y_g p + w) & m(z_g r + x_g p) & -m(y_g r - u) \\ 0 & 0 & 0 & -m(z_g p - v) & -m(z_g q + u) & m(x_g p + y_g q) \\ -m(y_g q + z_g r) & m(y_g p + w) & m(y_g p - v) & 0 & -I_{yz} q - I_{xz} q + I_z r & I_{yz} r + I_{xy} p - I_y q \\ m(x_g p - w) & -m(z_g r - x_g p) & m(z_g q + u) & I_{yz} q + I_{xz} p - I_z r & 0 & -I_{xz} r - I_{xy} q + I_X p \\ m(x_g r + v) & m(y_g r - u) & -m(x_g p + y_g q) & -I_{yz} r - I_{xy} p + I_y q & I_{xz} r + I_{xy} q - I_x p & 0 \end{bmatrix} \quad (\text{A.4})$$

$$a_1 = X_{\dot{u}} u + X_{\dot{v}} v + X_{\dot{w}} w + X_{\dot{p}} p + X_{\dot{q}} q + X_{\dot{r}} r \quad (\text{A.5})$$

$$a_2 = Y_{\dot{u}} u + Y_{\dot{v}} v + Y_{\dot{w}} w + Y_{\dot{p}} p + Y_{\dot{q}} q + Y_{\dot{r}} r \quad (\text{A.6})$$

$$a_3 = Z_{\dot{u}} u + Z_{\dot{v}} v + Z_{\dot{w}} w + Z_{\dot{p}} p + Z_{\dot{q}} q + Z_{\dot{r}} r \quad (\text{A.7})$$

$$b_1 = K_{\dot{u}}u + K_{\dot{v}}v + K_{\dot{w}}w + K_{\dot{p}}p + K_{\dot{q}}q + K_{\dot{r}}r \quad (\text{A.8})$$

$$b_2 = M_{\dot{u}}u + M_{\dot{v}}v + M_{\dot{w}}w + M_{\dot{p}}p + M_{\dot{q}}q + M_{\dot{r}}r \quad (\text{A.9})$$

$$b_3 = N_{\dot{u}}u + N_{\dot{v}}v + N_{\dot{w}}w + N_{\dot{p}}p + N_{\dot{q}}q + N_{\dot{r}}r \quad (\text{A.10})$$

Table A.2: Sail Angle Look-up Table

Apparent Wind Angle	Sail Angle	Apparent Wind Angle	Sail Angle
0	-90	1	-89
2	-89	3	-88
4	-89	5	-87
6	-88	7	-86
8	-87	9	-85
10	-86	11	-84
12	-85	13	-83
14	-84	15	-82
16	-83	17	-81
18	-82	19	-80
20	-81	21	-79
22	-80	23	-78
24	-79	25	-77
26	-78	27	-76
28	-77	29	-75
30	-76	31	-74
...
330	75	331	76
332	76	333	77
334	77	335	78
336	78	337	79
338	79	339	80
340	80	341	81
342	81	343	82
344	82	345	83
346	83	347	84
348	84	349	85
350	85	351	86
352	86	353	87
354	87	355	88
356	88	357	89
358	89	359	90

A.3. Ocean Vessel Parameters

Table A.3: Parameters of Container Vessel

Parameter Definition	Parameter Value	Parameter Definition	Parameter value
m	0.00792	N_v	-0.0038545
X_{uu}	-0.0004226	N_r	-0.00222
X_{vr}	-0.00311	N_p	0.000213
X_{rr}	0.0002	N_ϕ	-0.0001424
$X_{\phi\phi}$	-0.0002	N_{vvv}	0.001492
X_{vv}	-0.00386	N_{rrr}	-0.00229
K_v	0.0003026	N_{vvr}	-0.0424
K_r	-0.000063	N_{vrr}	0.00156
K_p	-0.0000075	$N_{vv\phi}$	-0.019058
K_ϕ	-0.000021	$N_{v\phi\phi}$	-0.0053766
K_{vvv}	0.002843	$N_{rr\phi}$	-0.0038592
K_{rrr}	-0.0000462	$N_{r\phi\phi}$	0.0024195
K_{vvr}	-0.000588	$Y_{r\phi\phi}$	-0.001368
K_{vrr}	0.0010565	$Y_{rr\phi}$	0.009325
$K_{vv\phi}$	-0.0012012	$Y_{v\phi\phi}$	0.00304
$K_{v\phi\phi}$	-0.0000793	$Y_{vv\phi}$	0.04605
$K_{rr\phi}$	-0.000243	Y_{vrr}	-0.0405
$K_{r\phi\phi}$	0.00003569	Y_{vvr}	0.0214
Y_v	-0.0116	Y_{rrr}	0.00177
Y_r	0.00242	Y_{vvv}	-0.109
Y_p	0	Y_ϕ	-0.000063

A.4. Heading Controller

A.5. Matlab Code

A.5.1. Sail Controller

```

1 %% Sail Controller
2 function y = sailController(u)
3 % Convert apparent wind angle to the range 0 -> 360
4 if(u<360)
5     % Sail control equation implementation
6     if(u<180)
7         y = -0.5*u;
8     else
9         y=-0.5*u+180;
10    end
11 else
12     % Sail control equation implementation
13     u=u-360;
14     if(u<180)
15         y = -0.5*u;
16     else
17         y=-0.5*u+180;
18     end
19 end
20 end

```

A.5.2. Heading Controller

```

1 %% Heading Controller
2 function [y,tack,tackDistance,tackLeft,tackRight] = ...
3     headingController(tackHeading,pathTrajectory,WA,
4 currentHeading,wayPointHeading)
5 % Change wind angle to follow same convection as waypoint ...
6     heading, ranges from -180 -> 180
5 if WA<-180
6     WA = WA + 360;
7 end

```

```
8 if WA>180
9     WA = WA - 360;
10 end
11
12 % Determine the state of sailboat
13 if      abs(wayPointHeading-WA)<45
14     % State 1: Tacking
15         % Initialize tacking right
16         if(abs(tackHeading)<15)
17             % Set tack angle to the right
18             y=WA+45;
19             % State value
20             tack=0.5;
21             % Tack Distance D
22             tackDistance=400;
23             % Tack Right Angle
24             tackRight=WA+45;
25             % Tack Left Angle
26             tackLeft=WA-45;
27     else
28         % Reference heading
29         y=tackHeading-currentHeading;
30         % State value
31         tack=1;
32         % Tack Right Angle
33         tackRight=WA+45;
34         % Tack Left Angle
35         tackLeft=WA-45;
36         % Tack Distance D
37         tackDistance=400;
38     end
39 % State 0: Normal Sail
40 else
41     % State Value
42     tack=0;
43     % Reference Heading
44     y=pathTrajectory;
45     % Reset tacking distance
46     tackDistance=100;
```

```

47 % Reset Right tack angle
48 tackRight=45;
49 % Reset Left tack angle
50 tackLeft=-45;
51 end

```

A.5.3. Waypoint Controller

```

1 %% Heading Controller
2 function [nextState,currentWaypoint] = ...
   wayPointController(R,currentState)
3 % Waypoint switching
4 % With each added waypoint another case should be added
5 switch currentState
6     % First waypoint
7     case 0
8         % Check if waypoint lies within radius of acceptance
9         if(R==1)
10            % Switch current Waypoint to next waypoint
11            nextState=1;
12            currentWaypoint=1;
13            % Waypoint not reached
14        else
15            nextState=0;
16            currentWaypoint=0;
17        end
18    case 1
19        % Check if waypoint lies within radius of acceptance
20        if(R==1)
21            % Switch current Waypoint to next waypoint
22            nextState=2;
23            currentWaypoint=2;
24            % Waypoint not reached
25        else
26            nextState=currentState;
27            currentWaypoint=1;
28        end

```

```
29 case 2
30     % Check if waypoint lies within radius of acceptance
31     if(R==1)
32         % Switch current Waypoint to next waypoint
33         nextState=3;
34         currentWaypoint=3;
35     % Waypoint not reached
36     else
37         nextState=currentState;
38         currentWaypoint=2;
39     end
40 case 3
41     % Check if waypoint lies within radius of acceptance
42     if(R==1)
43         % Switch current Waypoint to next waypoint
44         nextState=4;
45         currentWaypoint=4;
46     % Waypoint not reached
47     else
48         nextState=currentState;
49         currentWaypoint=3;
50     end
51 case 4
52     % Check if waypoint lies within radius of acceptance
53     if(R==1)
54         % Switch current Waypoint to next waypoint
55         nextState=0;
56         currentWaypoint=0;
57     % Waypoint not reached
58     else
59         nextState=currentState;
60         currentWaypoint=4;
61     end
62 otherwise
63     nextState=0;
64     currentWaypoint=0;
65 end
```

A.5.4. Waypoint Construction

```

1 %% Waypoints
2 Waypoints.x = [x0, x1, x2, x3];
3 Waypoints.y = [y0, y1, y2, y3];
4 Waypoints.R = [R0, R1, R2, R3];
5 waypoint = 0; % Starting Waypoint
6 Waypoints.number= [0,1,2,3];

```

A.6. Sailboat Specifications

Table A.4: Specifications of Dragonflite 95 Sailboat

Parameter Definition	Parameter Value
Length	950 mm
Width	125 mm
Rig Height	105 mm
Overall Height	1470 mm
RTR Total Weight	2000 g
Sail Area (Mainsail)	2314 cm ²
Sail Area (Jib)	1422 cm ²
Sail Area (Overall)	3736 cm ²
Hull Material	Molded plastic with painted finish and logo sticker

A.7. Pixhawk 6C Technical Specifications

Processors and Sensors

- FMU Processor: STM32H743
 - 32 Bit Arm® Cortex®-M7, 480MHz, 2MB memory, 1MB SRAM
- IO Processor: STM32F103
 - 32 Bit Arm® Cortex®-M3, 72MHz, 64KB SRAM
- On-board sensors
 - Accel/Gyro: ICM-42688-P
 - Accel/Gyro: BMI055
 - Mag: IST8310
 - Barometer: MS5611

Electrical data

- Voltage Ratings:
 - Voltage Ratings:
 - USB Power Input: 4.75-5.25V
 - Servo Rail Input: 0-36V
- Servo Rail Input: 0-36V
 - TELEM1 Max output current limiter: 1.5A
 - All other port combined output current limiter: 1.5A

Mechanical data

- Dimensions: 84.8 * 44 * 12.4 mm
- Weight: 59.3g

Interfaces

- 16- PWM servo outputs (8 from IO, 8 from FMU)
 - TELEM1 - Full flow control, separate 1.5A current limit
 - TELEM2 - Full flow control

- TELEM3
- 2 GPS ports
 - GPS1 - Full GPS port (GPS plus safety switch)
 - GPS2 - Basic GPS port
- 1 I2C port
 - Supports dedicated I2C calibration EEPROM located on sensor module
- 2 CAN Buses
 - CAN Bus has individual silent controls or ESC RX-MUX control
- 2 Debug ports:
 - FMU Debug
 - I/O Debug
- Dedicated R/C input for Spektrum / DSM and S.BUS, CPPM, analog / PWM RSSIs
- 2 Power input ports (Analog)
- Other Characteristics:
 - Operating and storage temperature: -40 to 85°C

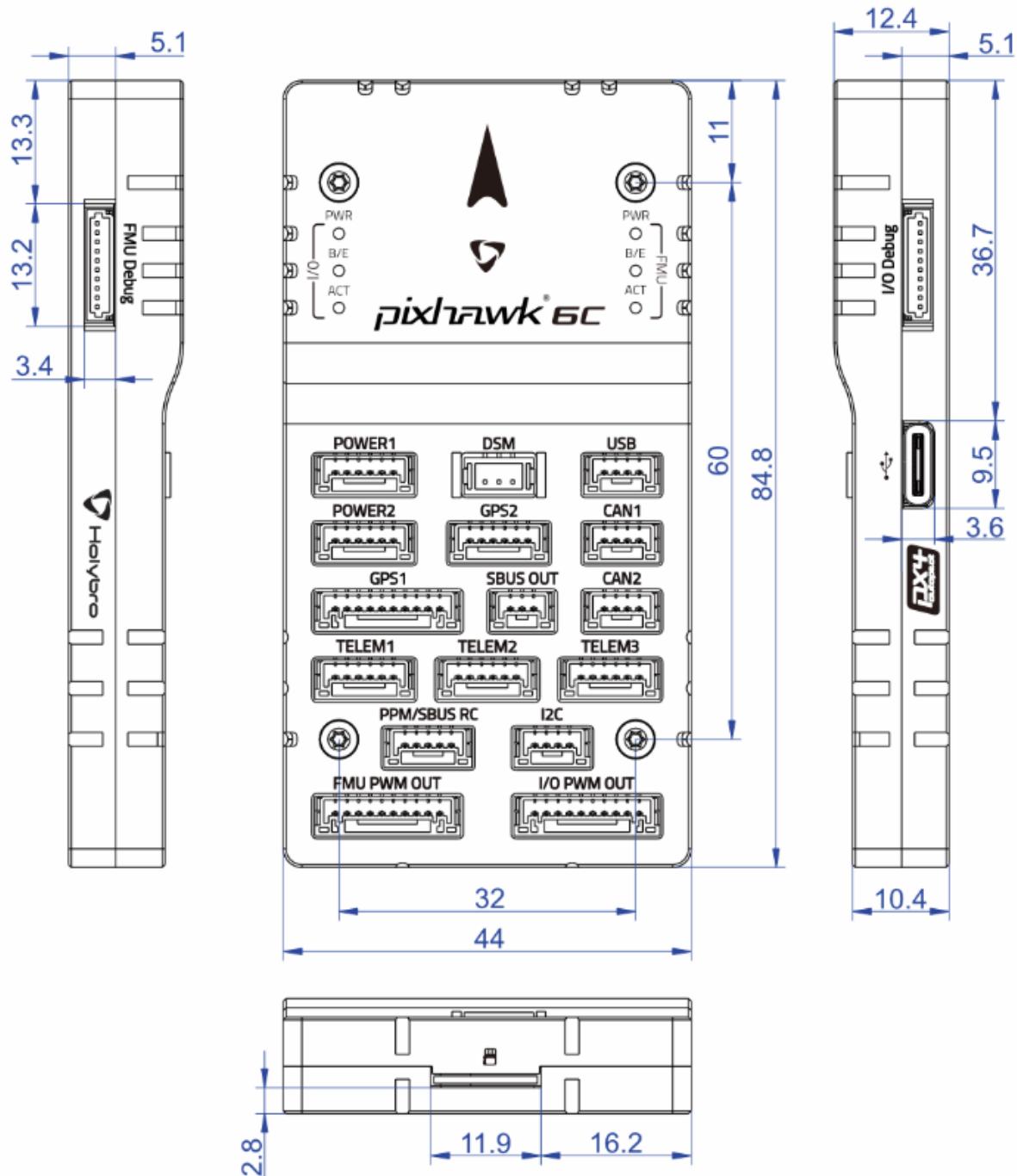


Figure A.1: Pixhawk 6C Dimensions

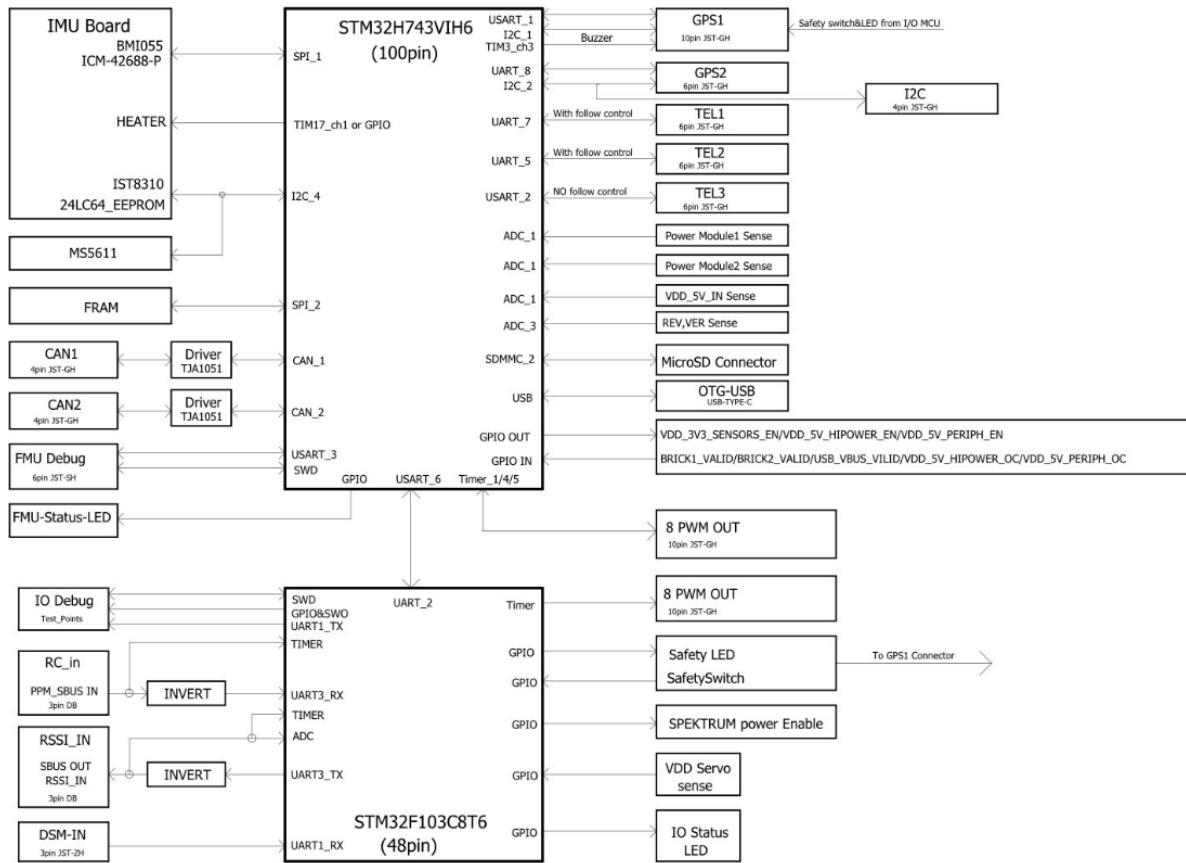


Figure A.2: Pixhawk 6C Pinout Diagram

A.8. M8N GPS Technical and Features Specifications

Features and Specifications:

- Ublox Neo-M8N module
- IST8310 compass
- Industry leading -167 dBm navigation sensitivity
- Cold starts: 26s
- LNA MAX2659ELT+
- 25 x 25 x 4 mm ceramic patch antenna
- Rechargeable Farah capacitance
- Low noise 3.3V regulator
- Current consumption: less than 150mA @ 5V
- Fix indicator LEDs

- Protective case
- Cable Length: 26cm
- Diameter 50mm total size, 32 grams with case

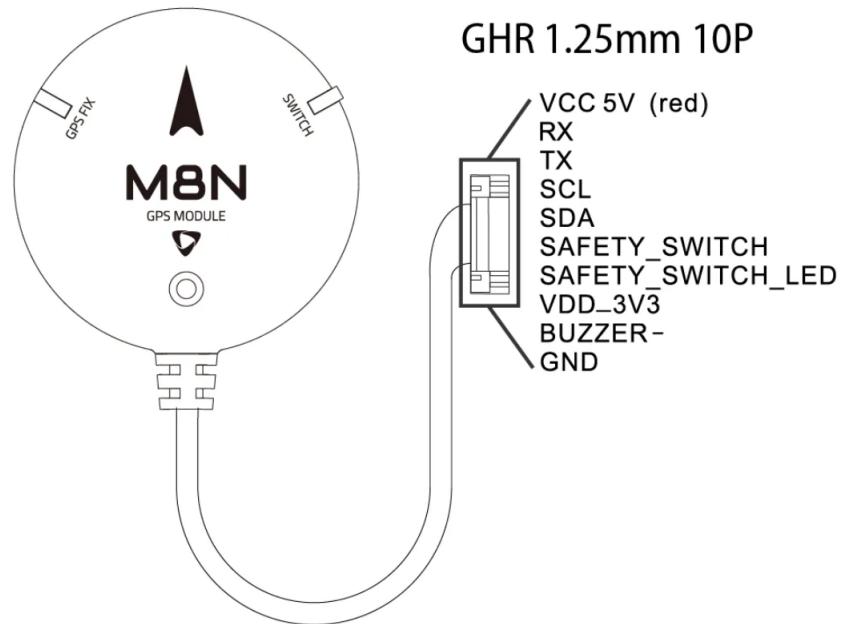


Figure A.3: M8N GPS pinout diagram

A.9. Ardupilots Sailboat Configuration Parameters

Table A.5: Sailboat Configuration Parameters

Parameter	Definition
SAIL_ENABLE	This enables Sailboat functionality
SAIL_ANGLE_MIN	Mainsheet tight, angle between centerline and boom
SAIL_ANGLE_MAX	Mainsheet loose, angle between centerline and boom
SAIL_ANGLE_IDEAL	Ideal angle between sail and apparent wind
SAIL_WNDSPD_MIN	Sailboat minimum wind speed to continue sail in, at lower wind speeds the sailboat will motor if one is fitted
SAIL_NO_GO_ANGLE	The typical closest angle to the wind the vehicle will sail at. The vehicle will sail at this angle when going upwind
SAIL_XTRACK_MAX	The sail boat will tack when it reaches this cross track error, defines a corridor of 2 times this value wide, 0 disables
PIVOT_TURN_RATE	Desired pivot turn rate in deg/s.
SAIL_LOIT_RADIUS	When in sailing modes the vehicle will keep moving within this loiter radius
SAIL_HELL_MAX	When in auto sail trim modes the heel will be limited to this value using PID control
ATC_SAIL_P	Sail Heel control P gain for sailboats. Converts the error between the desired heel angle (in radians) and actual heel to a main sail output (in the range -1 to +1)
ATC_SAIL_D	Sail Heel control D gain. Compensates for short-term change in desired heel angle vs actual
ATC_SAIL_I	Sail Heel control P gain for sailboats. Converts the error between the desired heel angle (in radians) and actual heel to a main sail output (in the range -1 to +1)
ATC_SAIL_IMAX	Sail Heel control I gain maximum. Constrains the maximum I term contribution to the main sail output (range -1 to +1)
SIM_SAIL_TYPE	Sailboat simulation sail type
ATC_SAIL_FF	Sail Heel control feed forward
ATC_SAIL_FILT	Sail Heel control input filter. Lower values reduce noise but add delay.

Table A.5: Sailboat Configuration Parameters

Parameter	Definition
SAIL_ENABLE	This enables Sailboat functionality
SAIL_ANGLE_MIN	Mainsheet tight, angle between centerline and boom
SAIL_ANGLE_MAX	Mainsheet loose, angle between centerline and boom
SAIL_ANGLE_IDEAL	Ideal angle between sail and apparent wind
SAIL_WNDSPD_MIN	Sailboat minimum wind speed to continue sail in, at lower wind speeds the sailboat will motor if one is fitted
SAIL_NO_GO_ANGLE	The typical closest angle to the wind the vehicle will sail at. The vehicle will sail at this angle when going upwind
SAIL_XTRACK_MAX	The sail boat will tack when it reaches this cross track error, defines a corridor of 2 times this value wide, 0 disables
PIVOT_TURN_RATE	Desired pivot turn rate in deg/s.
ATC_SAIL_FLTT	Target filter frequency in Hz
ATC_SAIL_FLTE	Error filter frequency in Hz
ATC_SAIL_FLTD	Derivative filter frequency in Hz
ATC_SAIL_SMAX	Sets an upper limit on the slew rate produced by the combined P and D gains

A.10. Copter Modes Architecture

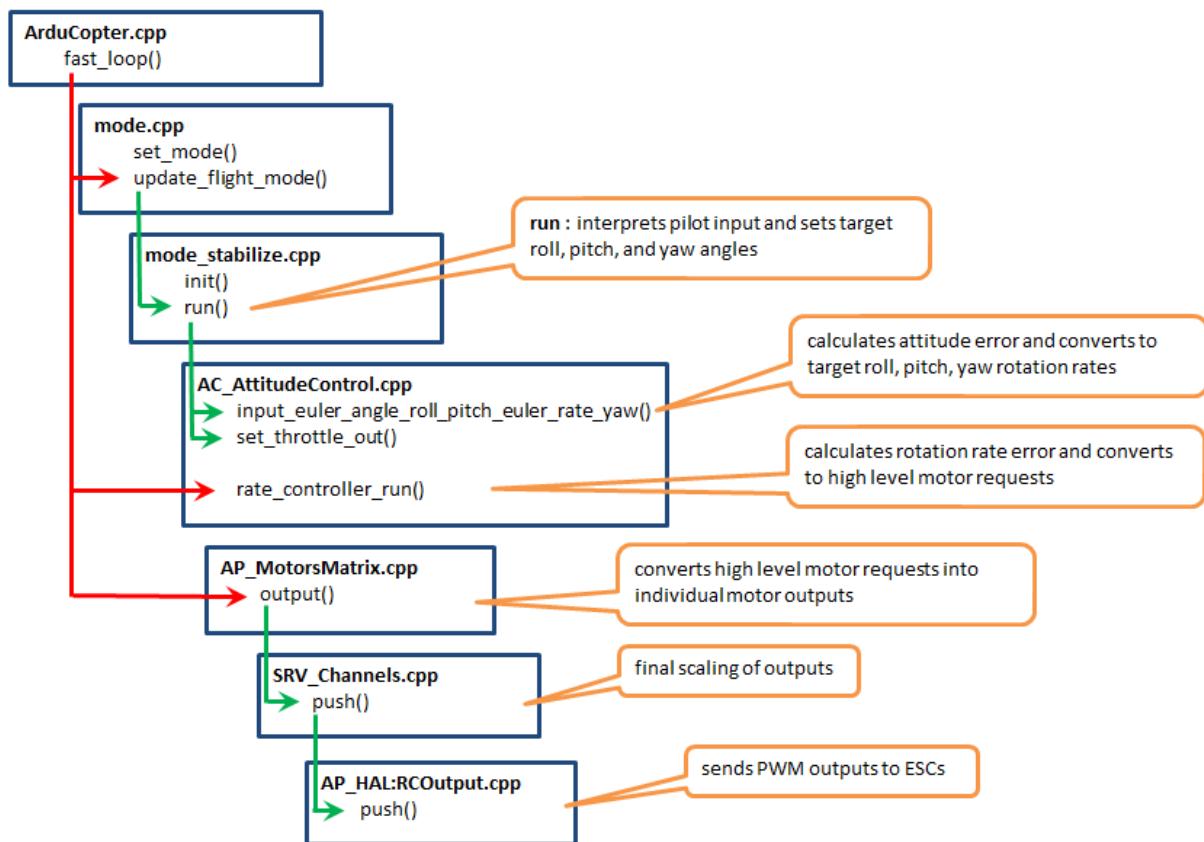


Figure A.4: Manual Flight Mode

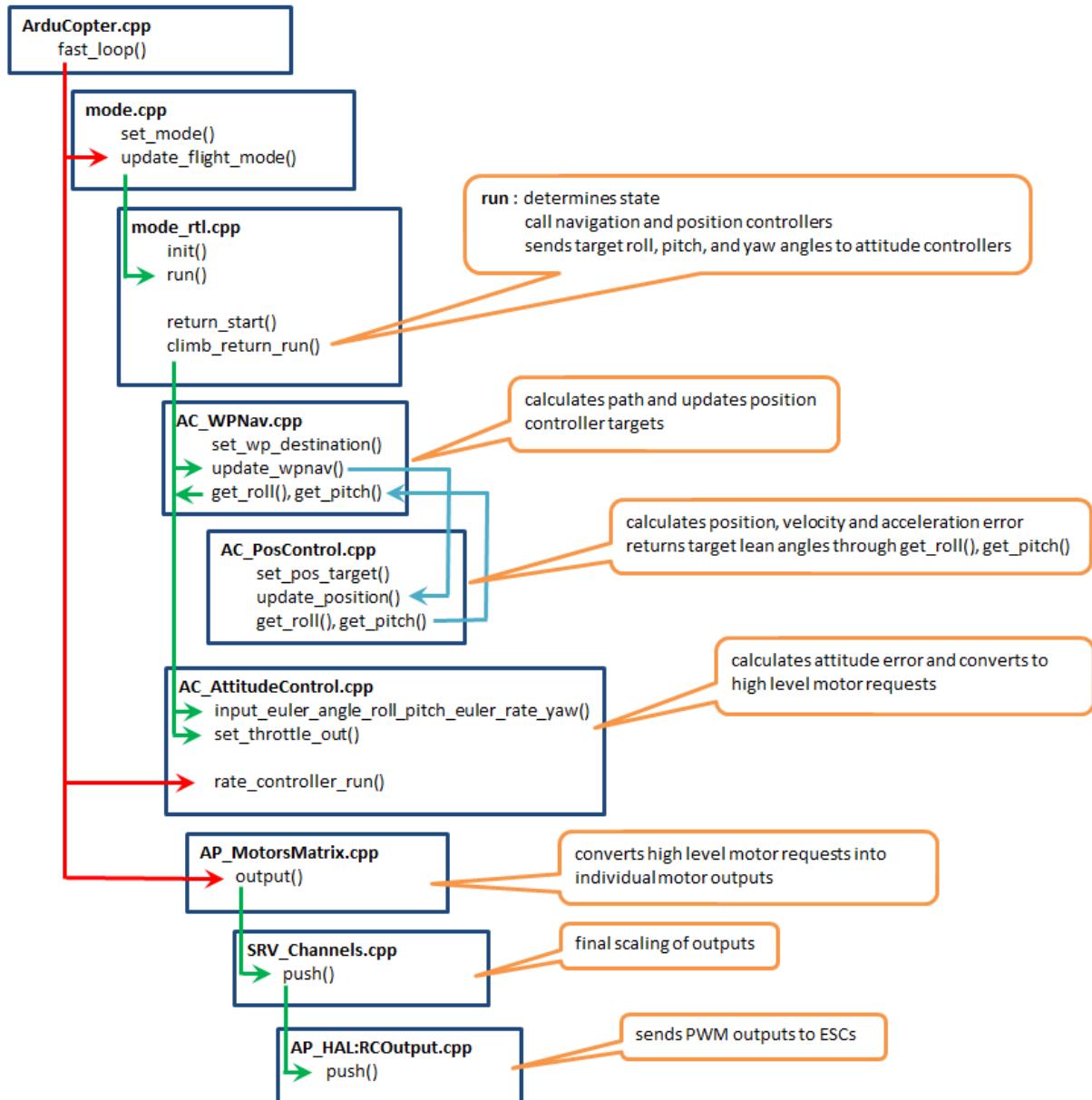


Figure A.5: Autonomous Flight Mode

A.11. Creating Custom Log Messages

When creating a new log message first, we firstly need to create a *struct* to hold the message declarations. The name *log_test* is given to the structure containing the information that is being sent through log messages. Every log message starts the same with *LOG_PACKET_HEADER* and *uint64_t time_us*. The rest of the structure is illustrated below

Listing A.1: Structure Definition

```

1 struct PACKED log_Test {
2     LOG_PACKET_HEADER;
3     uint64_t time_us;
4     float a_value;
5 }
```

The structure should be defined in the *Log.cpp* file of the **Rover** firmware. Also inside the *Log.cpp* file a function that sends the message structure defined above. The function assigns values to the message and sends it to the *logger*.

Listing A.2: Function Definition

```

1 void Rover::Log_Write_Test()
2 {
3     struct log_Test pkt = {
4         LOG_PACKET_HEADER_INIT(LOG_TEST_MSG),
5         time_us : AP_HAL::micros64(),
6         a_value : 1234
7     };
8     logger.WriteBlock(&pkt, sizeof(pkt));
9 }
```

The definition of the custom message should also be added to the *log_structure*. The definition is illustrated below, the definition for the letters is defined in *LogStructure.h* file.

Listing A.3: Message Definition

```

1 // @LoggerMessage: TEST
2 // @Description: Custom Log Message information
3 // @Field: TimeUS: Time since system startup
4 // @Field: Test: Test Var
5 { LOG_TEST_MSG, sizeof(log_Test),
6   "TEST", "Qf", "TimeUS,Test", "sm", "F0" },
7 };
```

The log message can be called in different ways and at different rates inside the *Rover.cpp* file(remember to also define function in *Rover.h*), below are the once a second loop.

Listing A.4: Log Message Call

```
1      /*
2       once a second events
3   */
4 void Rover::one_second_loop(void)
5 {
6     ...
7     Log_Write_Test();
8 }
```