

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. Sailing Principles	2
2.2. Recent Development in Autonomous Sailboats	4
2.3. Actuators for Sailboats	7
2.4. Sailboat Control Platforms	8
2.5. Autonomous Racing Sailboats	8
3. System Overview	10
3.1. Physical System	10
3.2. Marine Hardware	11
3.2.1. Autopilot Controller	12
3.2.2. GPS and Compass Sensor	13
3.2.3. Telemetry System	14
3.2.4. Wind Vane	14
3.3. Stationary Base-Station	15
3.3.1. Ground Control Software	15
3.4. Simulation Software	18
3.4.1. Matlab Simulator	18

3.4.2. ArduPilot Simulation	18
4. Ocean Vessel Dynamic Model	20
4.1. Reference Frames and Notations	20
4.1.1. Ocean Vessel Notation	21
4.1.2. Earth-Centered Reference Frames	21
4.1.3. Geographic Reference Frame	22
4.1.4. Sailboat Reference Frame	23
4.2. Rigid-Body Equations of Motion	25
4.2.1. Kinetics	25
4.2.2. Kinematics	26
4.3. Hydrodynamic and Hydrostatic Forces and Moments	27
4.3.1. Restoring Forces and Moments	28
4.4. Environmental Disturbances	30
4.4.1. Current-induced Forces and Moments	30
4.4.2. Wave-induced Forces and Moments	31
4.4.3. Wind-induced Forces and Moments	31
4.5. Dynamic Positioning Model	33
4.6. Simplified Maneuvering Model including Roll(4 DOF)	34
4.7. Modeling of a Keel, Rudder and Sail	35
4.7.1. Modeling of a Keel	35
4.7.2. Modeling of a Rudder	37
4.7.3. Modeling of a Sail	38
4.8. Sailboat Differential Equation	39
5. Sailing Control System Development	41
5.1. Sailing Control System Overview	41
5.1.1. Motion Control System Overview	42
5.2. Steering Controller Design	42
5.2.1. First Order Nomoto Model	42
5.2.2. Stability and Maneuverability	44
5.2.3. Acceleration Feedback	53
5.2.4. Proposed other Control Techniques	53
5.3. Sail Reference Angle	54
5.4. Roll Protection	56
5.5. Tacking Controller	56
5.6. Heading Controller	60

6. Guidance Control System Development	61
6.1. Guidance System	61
6.2. Straight path Generation based on Waypoints	63
6.3. Line-of-Sight Steering Law	64
6.4. Path-Following Controllers	67
6.5. Circle of Acceptance for Surface Vessel	69
7. Simulations	70
7.1. RSS Simulator	70
7.2. Motion Control System Simulation	72
7.2.1. Steering Controller Simulation	72
7.2.2. Sail Reference Angle Simulation	72
7.2.3. Tacking Controller Simulation	72
7.2.4. Heading Controller Simulation	72
7.3. Guidance System Simulation	72
7.3.1. Path Following Controller Simulation	72
7.3.2. Waypoint Navigation Simulation	72
7.3.3. Race Course Simulation	72
8. Ardupilot	73
8.1. Architecture	73
8.2. Rover Architecture	74
8.3. Sailboat Framework	78
8.3.1. The Shortcomings of Sailboat Framework	79
8.4. Ardupilots Control Strategies	80
8.4.1. Sail Controller	80
8.4.2. Rudder Controller	80
8.4.3. Roll Controller	81
8.4.4. Tacking	81
8.4.5. Waypoint Navigation	82
9. Results	83
10. Conclusion	84
Bibliography	85
A. Sailboat Specifications	88
A.1. Sailboat Specifications	88
A.2. Determination of Added Mass and Inertia Moment of the sailboat	89
A.3. Determination of Linear Damping Forces and Moments	92

A.4. Identification of Nomoto's First Order System	93
B. Additional Modelling Information	96
B.1. Modeling	96
B.1.1. Rigid-Body System Inertia Matrix M_{RB}	96
B.1.2. Rigid-Body Coriolis and Centripetal Matrix C_{RB}	96
C. Classical Control Systems Theory	98
C.1. Time-Domain Specifications	98
C.2. Root Locus Design	99
D. Additional Platform Development	102
D.1. Pixhawk 6C Technical Specifications	102
D.2. M8N GPS Technical and Features Specifications	105
D.3. Ardupilots Sailboat Configuration Parameters	107
D.4. Copter Modes Architecture	109
D.5. Creating Custom Log Messages	111

List of Figures

2.1. Main components of a sailboat	2
2.2. Sailing Zones	3
2.3. Diagram of sail forces	4
2.4. Aeolus, sailing boat of ETH Zurich	4
2.5. Avalon Sailboat	5
2.6. Iboat	6
2.7. UCT Sailboat	6
2.8. Sailboat used for modeling	7
2.9. Sailboat Surfaces	8
2.10. Sailboat Racing	9
3.1. Dragonflite 95 RC model sailboat	10
3.2. Block diagram overview of physical system	11
3.3. Block diagram of marine hardware	12
3.4. Pixhawk 6C	13
3.5. M8N GPS	13
3.6. Telemetry radio	14
3.7. High Level Communication	14
3.8. Caption	15
3.9. Block diagram of base station	15
3.10. Mission Planner GUI	16
3.11. An example of a mission consisting of waypoints and a fence	17
3.12. Overview of Matlab simulator	18
3.13. Architecture of Ardupilots SITL	19
4.1. The Earth-centered Earth-fixed (ECEF) frame and an Earth-centered inertial (ECI) frame	20
4.2. Motion variables for an ocean vessel	21
4.3. Body-fixed reference points	23
4.4. Sailboat roll reference angle	24
4.5. Sailboat sail and rudder reference angles	24
4.6. Transverse metacentric stability	28
4.7. Ocean vessel's heading angle	31
4.8. Wind angle on vessel	32

4.9.	Linear and quadratic damping and their speeds regimes	33
4.10.	Keel dimensions	36
4.11.	Lift and drag force on rudder	37
4.12.	Rudder dimensions	37
4.13.	Sail Dimensions	38
5.1.	Block diagram overview of Sailing Control System	41
5.2.	Pole placement	44
5.3.	XY - plot of straight-line stability	47
5.4.	Yaw	47
5.5.	Yaw Rate	47
5.6.	XY - plot of directional stability (critically damped)	48
5.7.	Yaw	48
5.8.	Yaw Rate	48
5.9.	Directional Stability (under damped)	48
5.10.	Positional Motion Stability	49
5.11.	PID rudder controller	49
5.12.	Root Locus of system	50
5.13.	Root Locus of plant with PD Controller	51
5.14.	Augmented system root Locus	52
5.15.	Step response of system	52
5.16.	Optimal sail angle for varying wind angle	55
5.17.	Dimensionless sail force	55
5.18.	State Diagram for tacking controller	57
5.19.	Direct tack maneuver	58
5.20.	Output and input of fuzzy logic controller	58
5.21.	Indirect tack calculation	59
5.22.	State diagram for heading controller	60
6.1.	Ground track between source waypoint and destination waypoint	61
6.2.	Cross-track error and in-track distance along track	62
6.3.	Guidance Axis System	62
6.4.	Straight lines and circular arc segments for waypoint guidance	64
6.5.	LOS guidance where the desired course angle x_d is chosen to point toward the LOS intersection point (x_{los}, y_{los})	65
6.6.	Circle of acceptance with constant radius R	66
6.7.	LOS guidance principle where the side slip angle B can be applied and compensated for by using integral action	68
6.8.	Definition of vector \bar{u} and \bar{v}	68

7.1.	Block diagram overview of RSS	71
7.2.	Overview of Simulink simulation	71
8.1.	High Level Architecture of Ardupilot Software	74
8.2.	Rover architecture	75
8.3.	Rover Controllers	76
8.4.	Rover navigation overview	76
8.5.	Formulas for L1 controllers	77
8.6.	Code Execution Flow for Sailboat in auto mode	79
8.7.	Ardupilots Rudder Controller	80
8.8.	Tacking	81
8.9.	Ardupilots Waypoint Navigation	82
A.1.	Vessel assumed as an ellipsoid	89
A.2.	Hull form with $H = 0.5$	91
A.3.	Notations used for the analysis of zig-zag maneuver	94
C.1.	Definition of time-domain specifications	98
C.2.	Basic closed-loop block diagram	100
D.1.	Pixhawk 6C Dimensions	104
D.2.	Pixhawk 6C Pinout Diagram	105
D.3.	M8N GPS pinout diagram	106
D.4.	Manual Flight Mode	109
D.5.	Autonomous Flight Mode	110

List of Tables

4.1. SNAME Notation for Ocean Vessels	21
5.1. Direct tacking controller state machine decisions	57
5.2. Indirect tacking controller state machine decisions	59
A.1. Sailboat Geometry	88
A.2. Sailboat Added Mass and Moment of Inertia	91
A.3. Sailboat Linear Damping Coefficients	93
A.4. Sailboat K and T Values	95
D.1. Sailboat Configuration Parameters	107
D.1. Sailboat Configuration Parameters	108

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

Sailboat Variables

p_1	Water friction coefficient
p_2	Water friction coefficient
p_3	Water angular friction coefficient
p_9	Roll friction coefficient
p_{10}	Length of the equivalent pendulum in roll motion
l_1	Length from CG to mast of sail
l_2	Length from CG to CoE of sail
l_3	Length from CG to CoE of keel
l_4	Length from CG to CoE of rudder
h_1	Height from CG to CoE of sail
h_2	Height from CG to CoE of keel
h_3	Height from CG to CoE of rudder

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
DP	Dynamic Positioning
CoE	Centre of Effort

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 autonomous sailboats. Sailboats can be used to explore area's of the ocean previously unexplored. Before the sailboat can be equipped with sensors to collect ocean data a reliable and robust control platform and system should be developed. This chapter will have a look at the recent development on sailboats and platforms currently available.

2.1. Sailing Principles

Sailboats are one of the oldest forms of transport and the first form of ocean transport. The skill to navigate the sea's has been acquired by the earliest human civilizations. The sailors would take note of the wind direction and then adjust the sail and sailboat heading appropriately. There are four main components of a sailboat, illustrated in Figure 2.1, the hull, rudder, keel and sail of the sailboat. The sail is usually separated into two parts, namely the main sail and the jib sail. The sail and the rudder are the only two actuators of a sailboat.

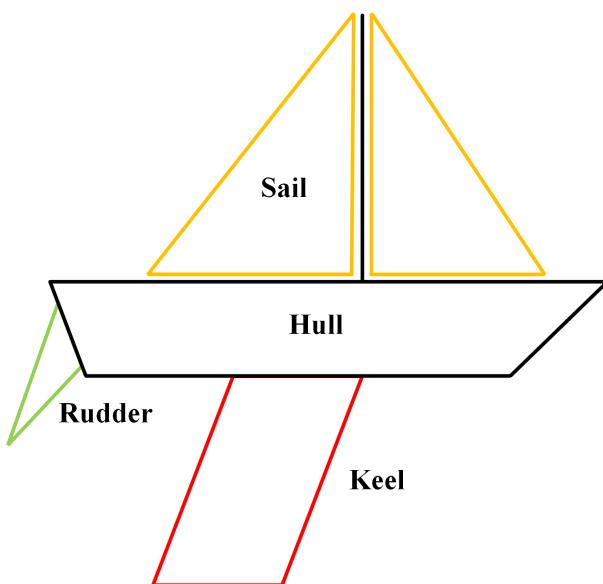


Figure 2.1: Main components of a sailboat

The sail is used to generate forward propulsion and the rudder is used to steer the sailboat in the desired heading. One limitation to take into account is the placement of the rudder compared to the sail. Because the sailboat is not equipped with a motor the rudder is only able to steer the sailboat when the sailboat has some forward velocity, also known as surge speed. The keel is used to balance the sailboat. The sail will cause unwanted force in the sway direction which is countered by the keel. The keel therefore generates equal and opposite forces caused by the sail to keep the sailboat sailing in a forward direction.

The sailboat is limited in its sailing direction with an inability to sail into the wind. This is due to the sail not being able to generate forward propulsion in the scenario. The area in which this occurs is known as the no sail zone or the no go zone. This is illustrated in Figure 2.2.

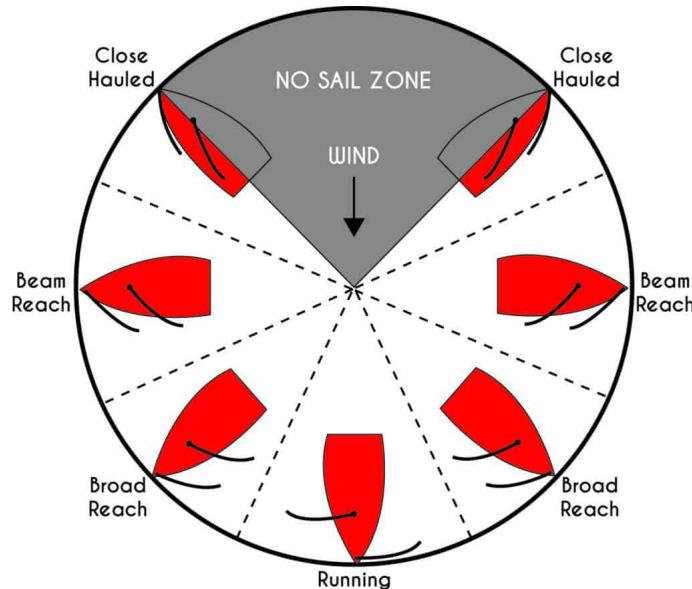


Figure 2.2: Sailing Zones

When sailing into the wind is when the sailboat is sail the fastest because it is utilising the lift generated by the sail compared to the drag of the sail which is used when sailing downwind. The forces generated by a sail is illustrated in Figure 2.3. A simple way to understand drag and lift is to remember drag is parallel to the apparent wind angle and lift is perpendicular to the drag

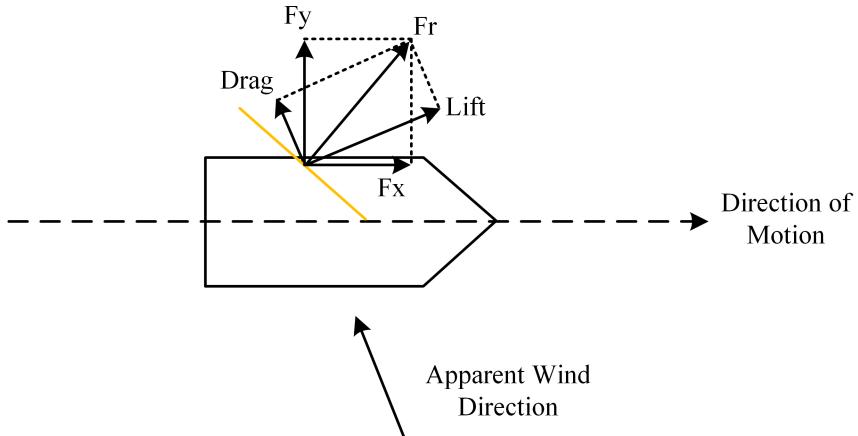


Figure 2.3: Diagram of sail forces

2.2. Recent Development in Autonomous Sailboats

Aeolus

The majority of development on autonomous sailboats has been restricted to small scale remote control sailboats. Some of the most notable sailboats development in recent years are, Aeolus from the university ETH Zurich [2], illustrated in Figure 2.4. Aeolus debuted in 2015 on the lake Zurich.



Figure 2.4: Aeolus, sailing boat of ETH Zurich

The platform demonstrated the ability of Aeolus to track a constant heading and perform tacking maneuvers. The rudder control was achieved by estimating a state-space model where, the states x_1 is the yaw and x_2 the yaw rate of the sailboat. The input of the

system is the rudder angle δ_r . The control platform, better known as an autopilot is a Pixhawk controller [3]. The Pixhawk comes equipped with the majority of the sensors required to implement control over the sailboat. The only extra sensor required is a wind sensor.

Avalon

The sailboat Avalon [4] is another sailboat developed in Zurich at the Federal Institute of Technology. Avalon is to date the most promising design and construction of a sailboat for research proposes with its most notable feature the use of two rudders. It is also one of the largest sailboats, being 3.95m, with an all in house developed hardware platform. The project however does not focus on the control of a sailboat but rather the mechanical design and construction of a sailboat.



Figure 2.5: Avalon Sailboat

Iboat

Iboat, illustrated in Figure 2.6, is one of the earliest developed sailboats. Iboat showcased how sailboats can be used for long-term offshore operation. The sailboat made use of a state machine based approach in sailing the sailboat. The three states are normal sailing, tacking and jibe. The project also showcases the importance of simulation in the development of an autonomous sailboat. The sailboat dynamics and the control systems were simulated in Matlab in a hardware in the loop type of simulation with sensor and actuation models.

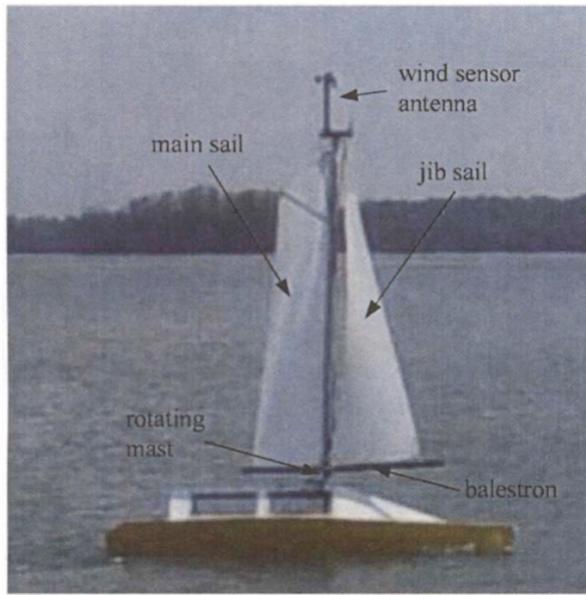


Figure 2.6: Iboat

University of Cape Town

The next sailboat being review is much closer to home. This is the sailboat developed by the University of Cape Town (UCT) [5]. The study focussed on building and modeling a fixed-wing sailboat. The modeling of the sailboat was done in 4 Degrees Of Freedom(DOF), which is the most popular method when modeling sailboats. The sailboat managed to sail with a fixed-wing which is a novel method of achieving forward propulsion. The study did not however touch on the subject of controlling a sailboat or the platform required to do so.



Figure 2.7: UCT Sailboat

Norwegian University of Science and Technology

An important part of sailboat control is to test the control systems in simulation on a realistic model of the sailboat. There is only however one of the biggest hurdles when designing sailboat control systems is accessing a simulation of a sailboat to test on. Currently the only simulations available were developed on Matlab and are limited to large sailboats usually in the class size of 7-10 meters. One of these simulations was developed at the Norwegian University of Science and Technology [6]. The study developed a 4-DOF model of the sailboat and simulated control. The study however was limited to only simulation and never tested on an actual sailboat, illustrated in Figure 2.8.



Figure 2.8: Sailboat used for modeling

Most of the research on sailboats lack on at least one of the following. A simulation platform for the specific envisioned sailboat. Control systems simulation on the sailboat and a reliable open-source autopilot with guaranteed future support. All of the above mentioned studies skipped on either one of these parts.

2.3. Actuators for Sailboats

It is important to know the available control surfaces available on sailboats and the controllability it gives you. For sailboats the actuators are simple and limited to two. Namely the rudder and the sail. The rudder is responsible for steering the sailboat in a desired heading and the sail is responsible for generating a forward propulsion. The sailboat surfaces are illustrated in Figure 2.9. The rudder surface are generally the same for more ships and if the design differs it does not change the control action required. But the sail can be implemented in two different methods. One being a flexible sail and the

other a fixed-wing. Flexible sails are much better research and older compared to fixed-wing sails. The fixed-wing does provided benefits being more robust and can withstand more harsh environment conditions compared to a flexible sail. Due to the fixed-wing option not being available for a small scale sailboat, it has to be built, a flexible sail option is favoured.



Figure 2.9: Sailboat Surfaces

2.4. Sailboat Control Platforms

2.5. Autonomous Racing Sailboats

Autonomous sailboat is still very much in its infancy with a hand full of sailboats developed. That said racing in autonomous sailboats is a topic barely researched, even with the huge fan base of remote controlled sailboats. The most notable racing competition is the international dragonflite 95 sailing competition [7]. The environment of racing with sailboats changes the goals for the autonomous sailboat. The control system now has to account for specific racing scenario's and rules.



Figure 2.10: Sailboat Racing

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 ... 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

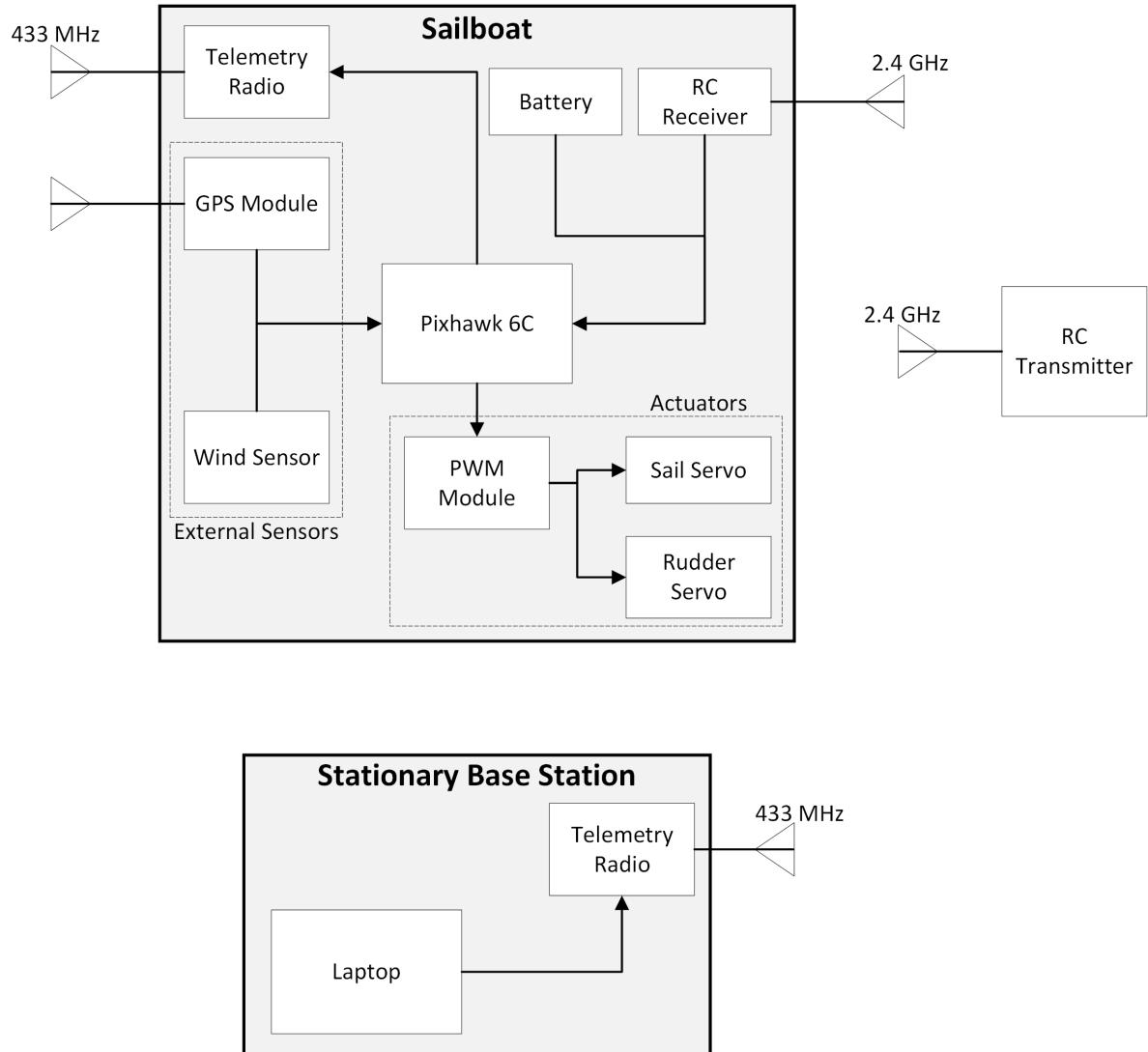


Figure 3.2: Block diagram overview of physical system

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

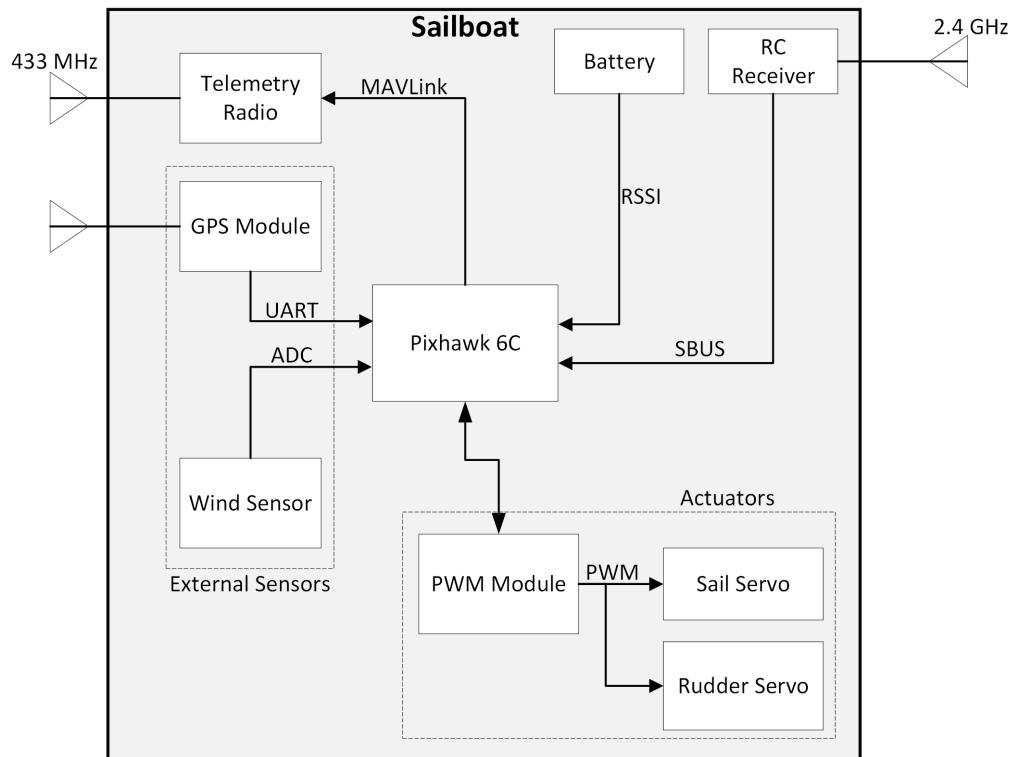


Figure 3.3: Block diagram of marine hardware

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 [3]. 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 D.1, the dimensions of the Pixhawk 6C is illustrated in Figure D.1 and the pinout diagram of the Pixhawk 6C is illustrated in Figure D.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 [10]. The sensor is illustrated in Figure 3.5. The technical and feature specifications are shown in Chapter D.2 and the pinout diagram is illustrated in Figure D.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 [11], 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.

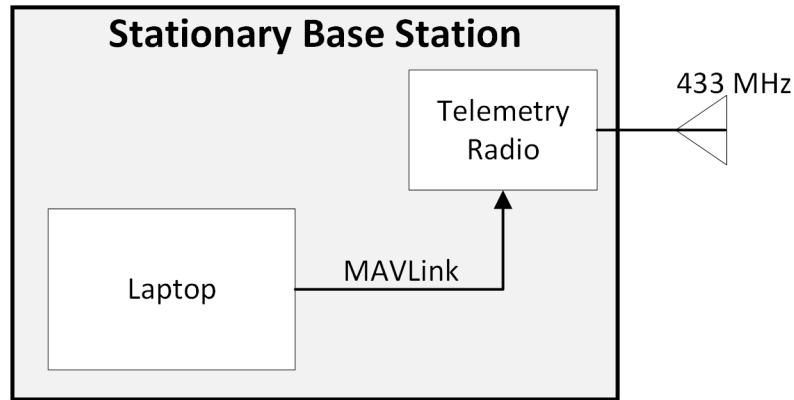


Figure 3.9: Block diagram of base station

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 [12]. 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.

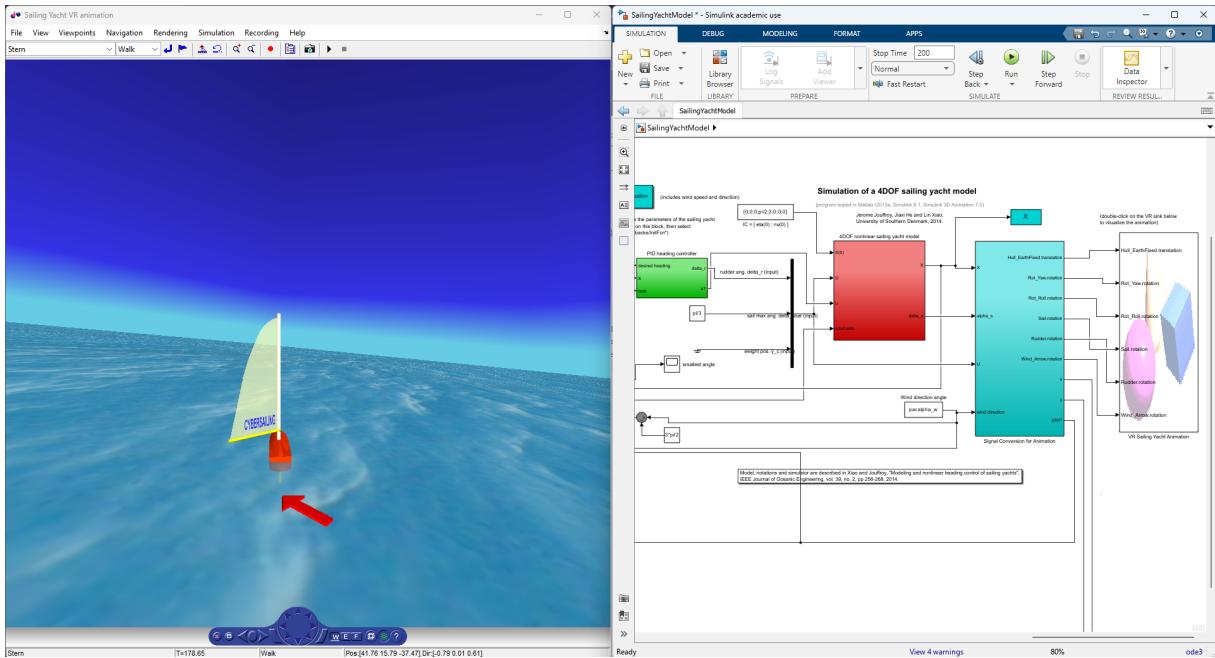


Figure 3.12: Overview of Matlab simulator

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.

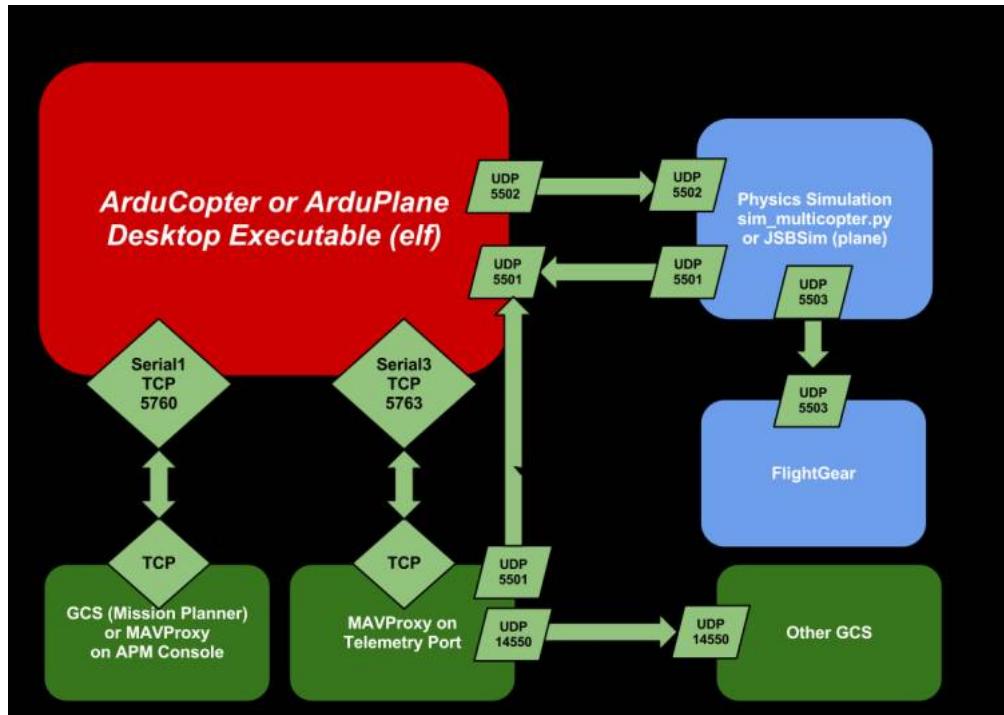


Figure 3.13: Architecture of Ardupilots SITL

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 [13]. 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.

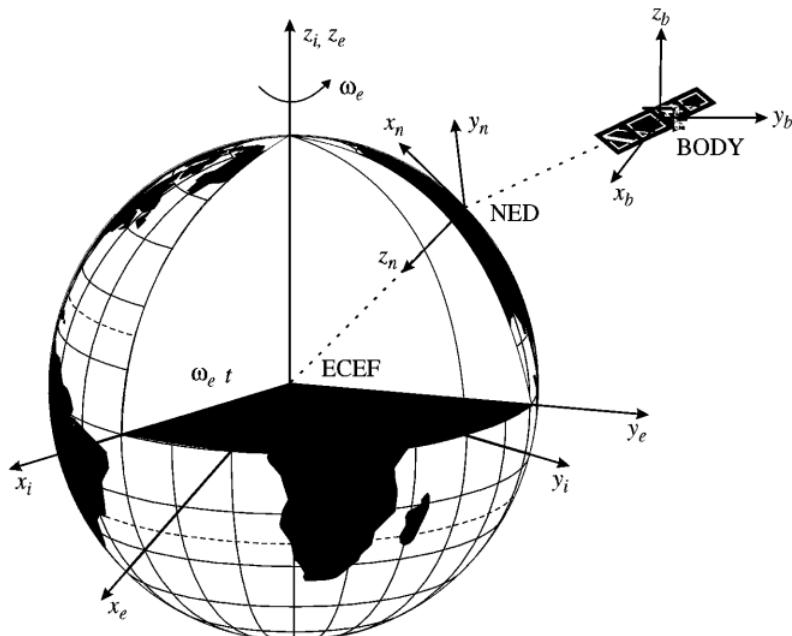


Figure 4.1: The Earth-centered Earth-fixed (ECEF) frame and an Earth-centered inertial (ECI) frame

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 [14]. Figure 4.2 illustrates the motion variables of an ocean vessel with the six independent coordinates.

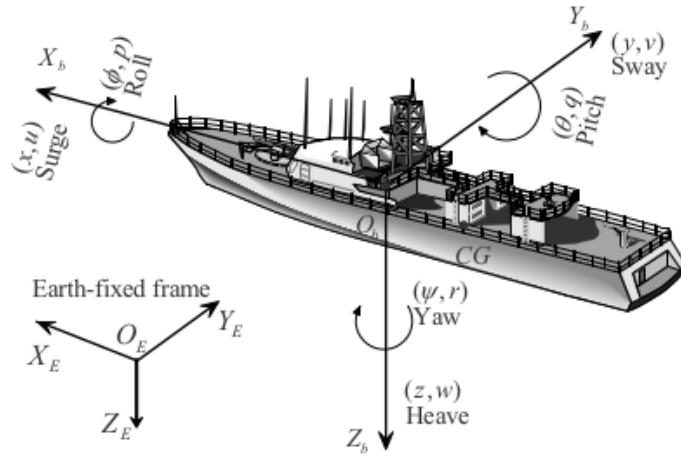


Figure 4.2: Motion variables for an ocean vessel

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. This point is referred to as *CO*, illustrated in Figure 4.3. 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)

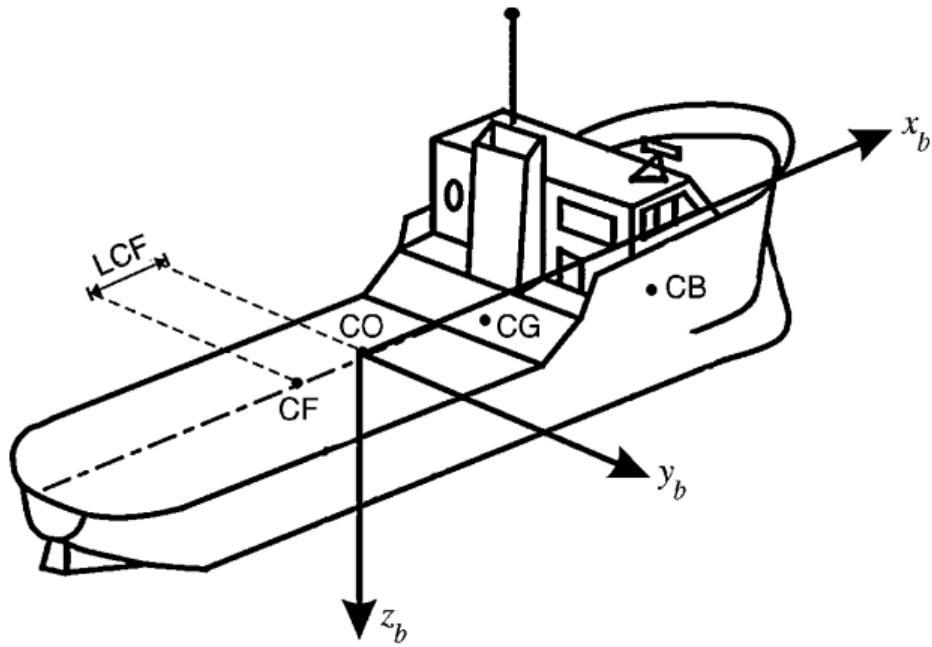


Figure 4.3: Body-fixed reference points

The following reference points are defined with respect to *CO*,

- **CG** - centre of gravity
- **CB** - centre of buoyancy
- **CF** - centre of flotation

The centre of flotation is the centroid of the water plane area A_{wp} in calm water. The vessel will roll and pitch around this point. Consequently this point can also be used to calculate the pitch and roll periods.

4.1.4. Sailboat Reference Frame

When modeling the sailboat the following assumptions are made [15]:

- The vessel will be assumed to be a rigid body with a 4-DOF namely surge, sway, yaw and roll.
- The mainsail and jib sail are modeled at the same sail angle and only the CoE of the main sail is used.
- The sail, rudder and keel are modeled as rigid foils and are computed independently with the roll angle of the sailboat considered.
- Added mass coefficients are modeled as constants

- The lift and drag coefficients are modeled as if the resultant force is applied at a single point on the rudder, keel and sail

The reference angles are illustrated in Figure 4.4 and Figure 4.5.

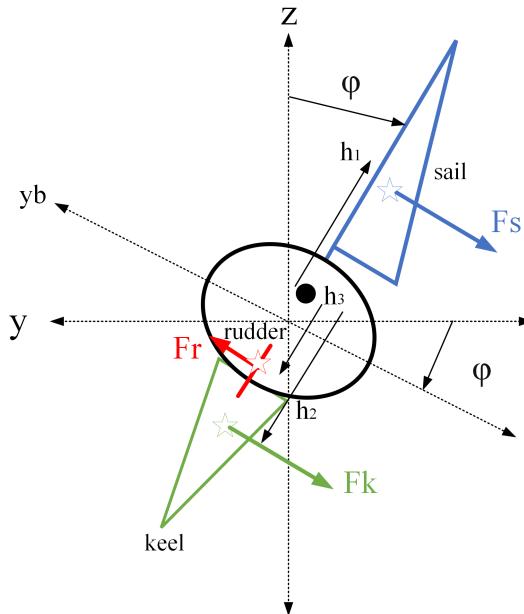


Figure 4.4: Sailboat roll reference angle

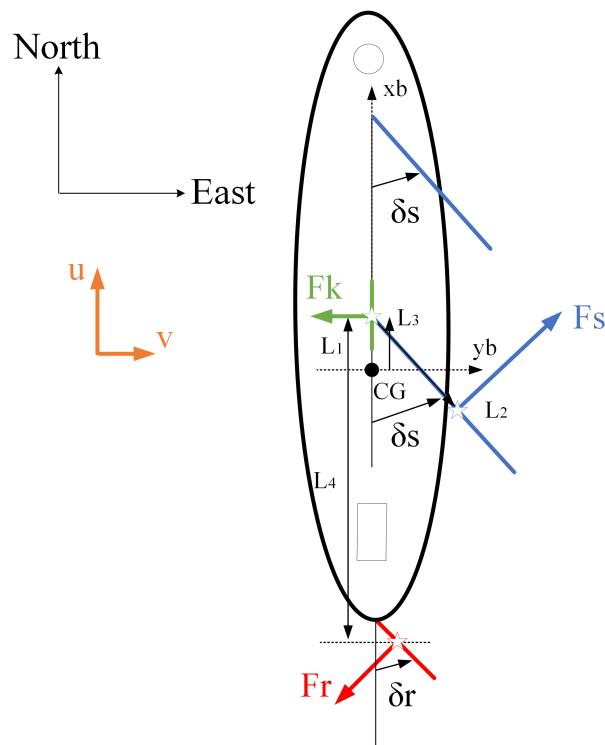


Figure 4.5: Sailboat sail and rudder reference angles

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 B.2 and B.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 linear velocity vectors is given by,

$$\mathbf{v}_1 = \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (4.11) \qquad \mathbf{v}_2 = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (4.12)$$

The position vectors is given by,

$$\mathbf{n}_1 = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (4.13) \qquad \mathbf{n}_2 = \begin{bmatrix} \Phi \\ \Theta \\ \Psi \end{bmatrix} \quad (4.14)$$

The \mathbf{J}_1 transformation matrix, where s denotes sin, c denotes cos and t denotes tan, is given by

$$\mathbf{J}_1(\mathbf{n}_2) = \begin{bmatrix} c\Psi c\Theta & -s\Psi c\Theta + s\Phi s\Theta c\Psi & s\Psi s\Phi + s\Theta c\Psi c\Phi \\ s\Psi c\Theta & c\Psi c\Phi + s\Phi s\Theta s\Psi & -c\Psi s\Phi + s\Theta s\Psi c\Phi \\ -s\Theta & s\Phi c\Theta & c\Phi c\Theta \end{bmatrix} \quad (4.15)$$

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

$$\mathbf{J}_2(\mathbf{n}_2) = \begin{bmatrix} 1 & -s\Phi t\Theta & c\Phi t\Theta \\ 0 & c\Phi & -s\Phi \\ 0 & s\Phi/c\Theta & c\Phi/c\Theta \end{bmatrix} \quad (4.16)$$

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 4.15 and Equation 4.16 results in the kinematics of an ocean vessel.

$$\begin{bmatrix} \dot{\mathbf{n}}_1 \\ \dot{\mathbf{n}}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{J}_1(\mathbf{n}_2) & 0_{3 \times 3} \\ 0_{3 \times 3} & \mathbf{J}_2(\mathbf{n}_2) \end{bmatrix} \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{bmatrix} = \dot{\mathbf{n}} = \mathbf{J}(\mathbf{n})\mathbf{v} \quad (4.17)$$

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,

$$\tau_{\mathbf{H}} = -\mathbf{M}_{\mathbf{A}}\dot{\mathbf{v}} - \mathbf{C}_{\mathbf{A}}(\mathbf{v})\mathbf{v} - \mathbf{D}(\mathbf{v})\mathbf{v} - \mathbf{g}(\mathbf{n}) \quad (4.18)$$

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,

$$\mathbf{M}_{\mathbf{A}} = \begin{bmatrix} X_{\dot{u}} & X_{\dot{v}} & X_{\dot{w}} & X_{\dot{p}} & X_{\dot{q}} & X_{\dot{r}} \\ Y_{\dot{u}} & Y_{\dot{v}} & Y_{\dot{w}} & Y_{\dot{p}} & Y_{\dot{q}} & Y_{\dot{r}} \\ Z_{\dot{u}} & Z_{\dot{v}} & Z_{\dot{w}} & Z_{\dot{p}} & Z_{\dot{q}} & Z_{\dot{r}} \\ K_{\dot{u}} & K_{\dot{v}} & K_{\dot{w}} & K_{\dot{p}} & K_{\dot{q}} & K_{\dot{r}} \\ M_{\dot{u}} & M_{\dot{v}} & M_{\dot{w}} & M_{\dot{p}} & M_{\dot{q}} & M_{\dot{r}} \\ N_{\dot{u}} & N_{\dot{v}} & N_{\dot{w}} & N_{\dot{p}} & N_{\dot{q}} & N_{\dot{r}} \end{bmatrix} \quad (4.19)$$

The hydrodynamic Coriolis and centripetal matrix is given below,

$$\mathbf{C}_{\mathbf{A}}(\mathbf{v}) = \begin{bmatrix} 0 & 0 & 0 & 0 & -a_3 & a_2 \\ 0 & 0 & 0 & a_3 & 0 & -a_1 \\ 0 & 0 & 0 & -a_2 & a_1 & 0 \\ 0 & -a_3 & a_2 & 0 & -b_3 & b_2 \\ a_3 & 0 & -a_1 & b_3 & 0 & -b_1 \\ -a_2 & a_1 & 0 & -b_2 & b_1 & 0 \end{bmatrix} \quad (4.20)$$

where a_1 , a_2 , a_3 , b_1 , b_2 and b_3 are defined in Equations

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,

$$\mathbf{D}(\mathbf{v}) = \mathbf{D} + \mathbf{D}_n(\mathbf{v}) \quad (4.21)$$

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

$$\mathbf{D} = - \begin{bmatrix} X_u & X_v & X_w & X_p & X_q & X_r \\ Y_u & Y_v & Y_w & Y_p & Y_q & Y_r \\ Z_u & Z_v & Z_w & Z_p & Z_q & Z_r \\ K_u & K_v & K_w & K_p & K_q & K_r \\ M_u & M_v & M_w & M_p & M_q & M_r \\ N_u & N_v & N_w & N_p & N_q & N_r \end{bmatrix} \quad (4.22)$$

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_{wp} denote the water plane area and

$$GM_T = \text{transverse metacentric height}(m) \quad (4.23)$$

$$GM_L = \text{longitudinal metacentric height}(m) \quad (4.24)$$

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 4.6.

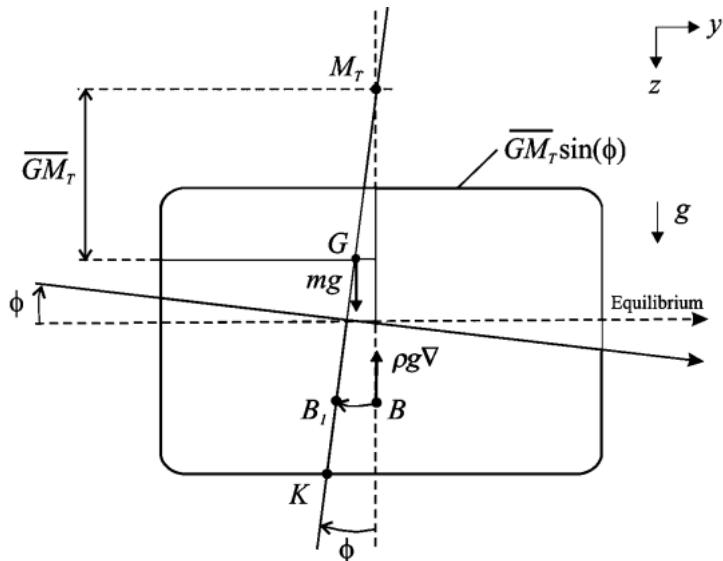


Figure 4.6: Transverse metacentric stability

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

$$mg = \rho g \nabla \quad (4.25)$$

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:

$$Z = mg - \rho g [\nabla + \delta \nabla(z)] = -\rho g \delta \nabla(z) \quad (4.26)$$

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

$$\delta \nabla(z) = \int_0^z A_{\omega\rho}(\delta) d\delta \quad (4.27)$$

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

$$Z \approx -\rho g A_{\omega\rho}(0) z \quad (4.28)$$

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

$$\delta f_r^b = -\rho g \begin{bmatrix} -s\Theta \\ c\Theta \ s\Phi \\ c\Theta \ c\Phi \end{bmatrix} \int_0^z A_{\omega\rho}(\delta) d\delta \quad (4.29)$$

From Figure 4.6 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,

$$r_r^b = \begin{bmatrix} -GM_L s\Theta \\ GM_T s\Phi \\ 0 \end{bmatrix} \quad (4.30)$$

$$f_r^b = -0\rho g \nabla \begin{bmatrix} -s\Theta \\ c\Theta \ s\Phi \\ c\Theta \ c\Phi \end{bmatrix} \quad (4.31)$$

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

$$m_r^b = r_r^b \times f_r^b = -\rho g \nabla \begin{bmatrix} GM_T s\Phi c\Theta c\Phi \\ GM_L s\Theta c\Theta c\Phi \\ (GM_T - GM_L c\Theta) s\Phi s\Theta \end{bmatrix} \quad (4.32)$$

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

$$\mathbf{g}(\mathbf{n}) = - \begin{bmatrix} \delta \mathbf{f}_r^b \\ \mathbf{m}_r^b \end{bmatrix} \quad (4.33)$$

$$\mathbf{g}(\mathbf{n}) = \begin{bmatrix} \rho g \int_0^z A_{\omega\rho}(\delta) d\delta s\Theta \\ \rho g \int_0^z A_{\omega\rho}(\delta) d\delta c\Theta s\Phi \\ \rho g \int_0^z A_{\omega\rho}(\delta) d\delta c\Theta c\Phi \\ \rho g \nabla GM_T s\Phi c\Theta c\Phi \\ \rho g \nabla GM_L s\Theta c\Theta c\Phi \\ \rho g \nabla (GM_T + -GM_L c\Theta) s\Phi s\Theta \end{bmatrix} \quad (4.34)$$

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.

$$\tau_E = \tau_E^{cu} + \tau_E^{wa} + \tau_E^{wi} \quad (4.35)$$

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

$$\tau_E^{cu} = (\mathbf{M}_{RB} + \mathbf{M}_A) \dot{\mathbf{v}}_c + \mathbf{C}(\mathbf{v}_r) \mathbf{v}_r - \mathbf{C}(\mathbf{v}) \mathbf{v} + \mathbf{D}(\mathbf{v}_r) \mathbf{v}_r - \mathbf{D}(\mathbf{v}) \mathbf{v} \quad (4.36)$$

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

$$\begin{bmatrix} u_c \\ v_c \\ w_c \end{bmatrix} = \mathbf{J}_1^T(\mathbf{n}_2) \begin{bmatrix} u_C^E \\ v_C^E \\ w_C^E \end{bmatrix} \quad (4.37)$$

4.4.2. Wave-induced Forces and Moments

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

$$\tau_E^{wa} = \begin{bmatrix} \sum_{i=1}^N \rho g BL T \cos(\beta) s_i(t) \\ \sum_{i=1}^N \rho g BL T \sin(\beta) s_i(t) \\ 0 \\ 0 \\ 0 \\ \sum_{i=1}^N \frac{1}{24} \rho g BL (L^2 - B^2) \sin(2\beta) s_i^2(t) \end{bmatrix} \quad (4.38)$$

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

$$s_i(t) = A_i \frac{2\pi}{\lambda_i} \sin(\omega_{ei} t + \phi_i) \quad (4.39)$$

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 .

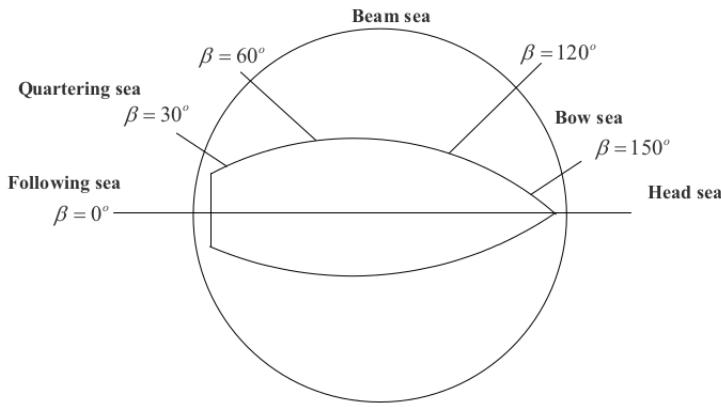


Figure 4.7: Ocean vessel's heading angle

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

$$\tau_E^{\omega i} = \begin{bmatrix} C_X(\gamma_\omega)A_{F\omega} \\ C_Y(\gamma_\omega)A_{L\omega} \\ C_Z(\gamma_\omega)A_{F\omega} \\ C_K(\gamma_\omega)A_{L\omega}H_{L\omega} \\ C_M(\gamma_\omega)A_{F\omega}H_{F\omega} \\ C_N(\gamma_\omega)A_{L\omega}L_{oa} \end{bmatrix} \quad (4.40)$$

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 4.8 and is given by

$$\gamma_\omega = \psi - \beta_\omega - \pi \quad (4.41)$$

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].

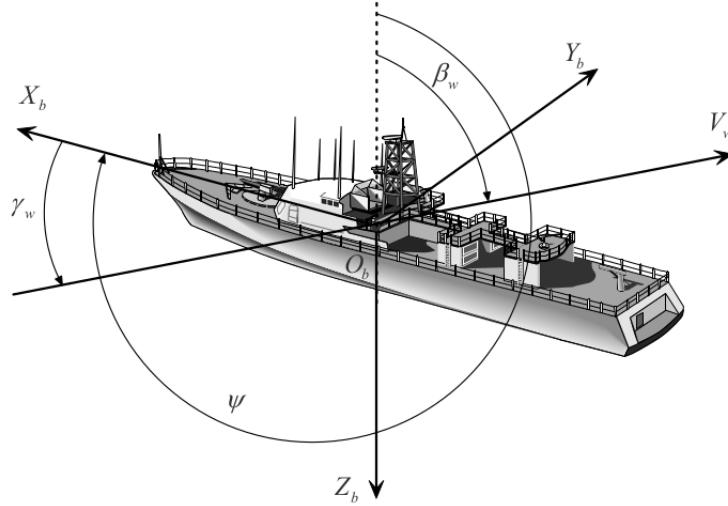


Figure 4.8: Wind angle on vessel

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

$$\tau_E^{\omega i} = \begin{bmatrix} C_X(\gamma_{r\omega})A_{F\omega} \\ C_Y(\gamma_{r\omega})A_{L\omega} \\ C_Z(\gamma_{r\omega})A_{F\omega} \\ C_K(\gamma_{r\omega})A_{L\omega}H_{L\omega} \\ C_M(\gamma_{r\omega})A_{F\omega}H_{F\omega} \\ C_N(\gamma_{r\omega})A_{L\omega}L_{oa} \end{bmatrix} \quad (4.42)$$

where

$$V_{r\omega} = \sqrt{u_{r\omega}^2 + v_{r\omega}^2} \quad (4.43)$$

$$\gamma_{r\omega} = -\arctan 2(v_{r\omega}, u_{r\omega}) \quad (4.44)$$

with

$$u_{r\omega} = u - V_\omega \cos(\beta_\omega - \psi) \quad (4.45)$$

$$v_{r\omega} = v - V_\omega \cos(\beta_\omega - \psi) \quad (4.46)$$

4.5. Dynamic Positioning Model

Dynamic Positioning (DP) model is used when ocean vessels are moving at low speeds. The approximate maximum speed where DP model is relevant is 2 m.s^{-1} , as illustrated in Figure 4.9. The nonlinear DP model is based on current coefficients and linear exponential damping that can be used for accurate simulation and prediction. As shown in [13] the quadratic damping can be neglected if appropriate compensation is performed for drift through integral action.

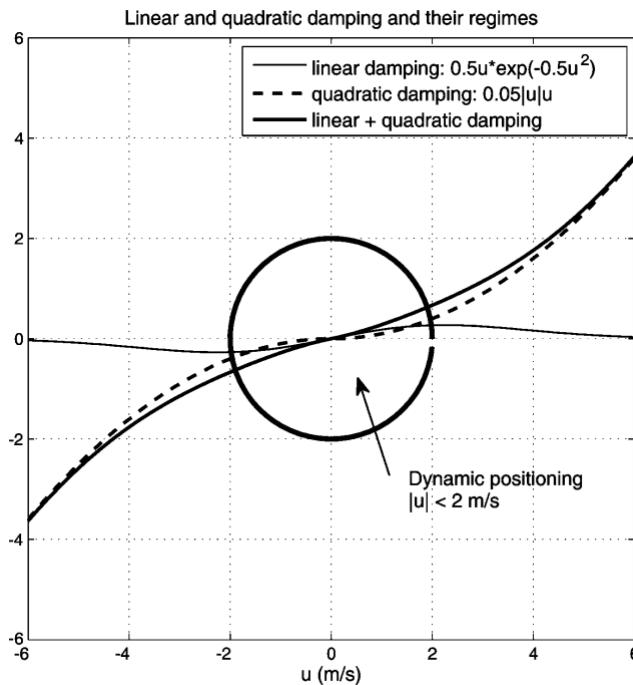


Figure 4.9: Linear and quadratic damping and their speeds regimes

The nonlinear vector representation is shown below,

$$\dot{\mathbf{n}} = \mathbf{R}(\Psi)\mathbf{v} \quad (4.47)$$

$$\mathbf{M}\dot{\mathbf{v}} + \mathbf{C}_{RB}(\mathbf{v})\mathbf{v} + \mathbf{N}(\mathbf{v}_r)\mathbf{v}_r = \boldsymbol{\tau} + \boldsymbol{\tau}_{wind} + \boldsymbol{\tau}_{wave} \quad (4.48)$$

where,

$$\mathbf{N}(\mathbf{v}_r)\mathbf{v}_r = \mathbf{C}_A(\mathbf{v}_r)\mathbf{v}_r + \mathbf{D}(\mathbf{v}_r)\mathbf{v}_r \quad (4.49)$$

The state vectors are $\mathbf{v} = [u, v, r]^T$ and $\mathbf{n} = [N, E, \Psi]^T$. The dynamics associated with the motion in heave, roll and pitch are neglected, that is $\omega = p = q = 0$. The rotation, mass and Coriolis matrices are,

$$\mathbf{R}(\Psi) = \begin{bmatrix} \cos(\Psi) & -\sin(\Psi) & 0 \\ \sin(\Psi) & \cos(\Psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.50)$$

$$\mathbf{M} = \begin{bmatrix} m - X_{\dot{u}} & 0 & 0 \\ 0 & m - Y_{\dot{v}} & mx_g - Y_{\dot{r}} \\ 0 & mx_g - Y_{\dot{r}} & I_z - N_{\dot{r}} \end{bmatrix} \quad (4.51)$$

$$\mathbf{C}_{RB}(\mathbf{v}) = \begin{bmatrix} 0 & 0 & -m(x_g r + v) \\ 0 & 0 & mu \\ m(x_g r + v) & -mu & 0 \end{bmatrix} \quad (4.52)$$

$$\mathbf{D}(\mathbf{v}_r)\mathbf{v}_r = \begin{bmatrix} -X_u & 0 & 0 \\ 0 & -Y_v & -Y_r \\ 0 & -N_v & -N_r \end{bmatrix} \quad (4.53)$$

In the DP model the surge is decoupled from sway and yaw and this is due to symmetry considerations of the system inertia matrix. This model is also the model that will be used when designing the rudder controller for the sailboat, the designed is discussed in Section 5.2.

4.6. Simplified Maneuvering Model including Roll(4 DOF)

The 3 DOF dynamic positioning equations can be extended 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,

$$\mathbf{M}\dot{\mathbf{v}} + \mathbf{C}_{RB}(\mathbf{v})\mathbf{v} + \mathbf{C}_A(\mathbf{v})\mathbf{v} + \mathbf{D}(\mathbf{v})\mathbf{v} + \mathbf{G}\mathbf{n} = \boldsymbol{\tau} + \boldsymbol{\tau}_{wind} + \boldsymbol{\tau}_{wave} + \boldsymbol{\tau}_{ocean} \quad (4.54)$$

where

$$\mathbf{M} = \begin{bmatrix} m - X_{\dot{u}} & 0 & 0 & 0 \\ 0 & m - Y_{\dot{v}} & -mz_g - Y_{\dot{p}} & mx_g - Y_{\dot{r}} \\ 0 & -mz_g - K_v & I_x - K_{\dot{p}} & 0 \\ 0 & mx_g - N_i & 0 & I_z - N_r \end{bmatrix} \quad (4.55)$$

$$\mathbf{C}_{RB}(\mathbf{v}) = \begin{bmatrix} 0 & 0 & mz_g r & -m(x_g r + v) \\ 0 & 0 & 0 & mu \\ -mz_g r & 0 & 0 & 0 \\ m(x_g r + v) & -mu & 0 & 0 \end{bmatrix} \quad (4.56)$$

$$\mathbf{C}_A(\mathbf{v}) = \begin{bmatrix} 0 & 0 & 0 & Y_i v \\ 0 & 0 & 0 & -X_{\dot{u}} u \\ 0 & 0 & 0 & Y_i v \\ -Y_i v & X_{\dot{u}} u & -Y_i v & 0 \end{bmatrix} \quad (4.57)$$

$$\mathbf{G} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -K_\phi & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (4.58)$$

This model is important in the design of a roll protection system for the sailboat. The main contributor to roll is the sail and wind angle. The affects of the sail, rudder and keel on the kinetic equations will also be modelled.

4.7. Modeling of a Keel, Rudder and Sail

The sailboat modeling refers to the forces and moments acting on the sailboat's specific added dimensions namely the extended keel, rudder and sail. The extra long keel acts as a counter force to the sail in the roll dimension. The rudders main function is to turn the sailboat in the desired heading and the sail is the sailboats form of propulsion. The modeling consist of combining multiple sources [20], [21], [22], [23], [24] and [22]. The keel, rudder and sail will only be modeled in 4-DOF as defined in Section 4.6.

4.7.1. Modeling of a Keel

When modeling the keel it is assumed that the keel can be modeled as foils. The drag and the lift of a foil can be calculated as,

$$F_{KL} = \frac{1}{2} \rho A C_L(\alpha) V^2 \quad (4.59)$$

$$F_{KD} = \frac{1}{2} \rho A C_D(\alpha) V^2 \quad (4.60)$$

where ρ is the density of the fluid, in the case of water, A is the area of the keel. The keel area is approximated and illustrated in Figure 4.10, V is the absolute velocity of the fluid relative to the keel and C_L and C_D is the respective lift and drag coefficients which are a function of the angle of attack α .

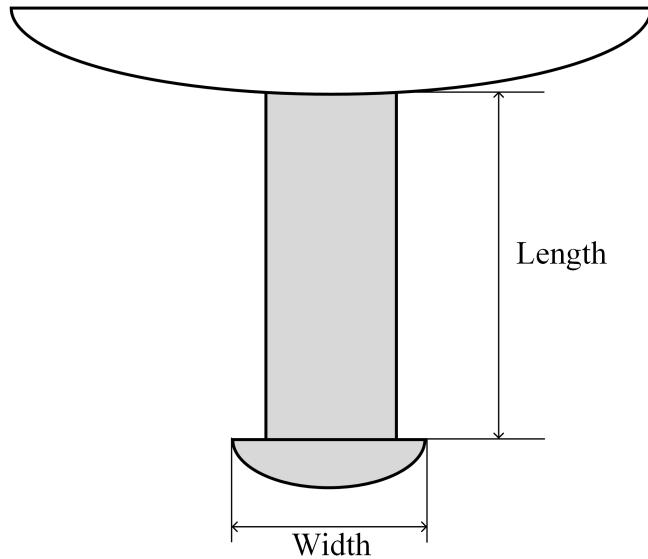


Figure 4.10: Keel dimensions

The lift and drag coefficients can be estimated by the following equations,

$$C_L = \frac{C_{L,2D}}{1 + \frac{2}{WL}} \alpha \quad (4.61)$$

$$C_D = \frac{C_L^2}{\pi WL} \quad (4.62)$$

where W and L is the width and length respectively of the keel. $C_{L,2D}$ is the two dimensional lift coefficient of a keel and can be approximated as $\frac{\pi}{90}$. To calculate the angle of angle of the water on the keel the relative velocity of the water needs to be calculated first. The relative velocity of the water on the keel is equal to,

$$V_{vk} = \begin{bmatrix} V_{vk_u} \\ V_{vk_v} \end{bmatrix} = - \begin{bmatrix} u_r \\ v_r \end{bmatrix} - \begin{bmatrix} 0 \\ -rl_3 - ph_2 \end{bmatrix} \quad (4.63)$$

The angle of the relative speed of the water is calculated as,

$$\beta_{vk} = \arctan V_{vk_u}, V_{vk_v} \quad (4.64)$$

The keel angle of attack is thus equal to,

$$\alpha_k = -\beta_{vk} + \pi \quad (4.65)$$

Now forces and moments generated by the keel can be calculated and is equal to,

$$K = \begin{bmatrix} F_{KL} \sin \beta_{vk} + F_{KD} \cos \beta_{vk} \\ (-F_{KL} \cos \beta_{vk} + F_{KD} \sin \beta_{vk}) \cos(\phi) \\ h_2 F_{KL} \cos \beta_{vk} - F_{KD} \sin \beta_{vk} \\ (l_3 F_{KL} \cos \beta_{vk} - F_{KD} \sin \beta_{vk}) \cos(\phi) \end{bmatrix} \quad (4.66)$$

where ϕ is the roll angle of the sailboat.

4.7.2. Modeling of a Rudder

The rudder of the sailboat is also assumed to be rigid foil. The lift and drag coefficient is calculated the same way as the in case of a keel. The rudder adds an extra layer of complexity due to it being an actuator and the rudder angle being changed constantly. The lift and drag force directions and angle of attack is illustrated in Figure 4.11.

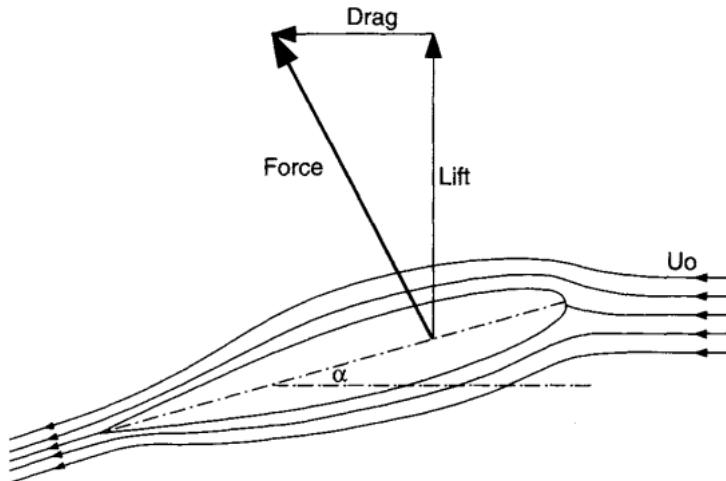


Figure 4.11: Lift and drag force on rudder

The length and width of the rudder is illustrated in Figure 4.12. The area of the rudder is not approximated seeing that it takes the shape of a perfect rectangle.

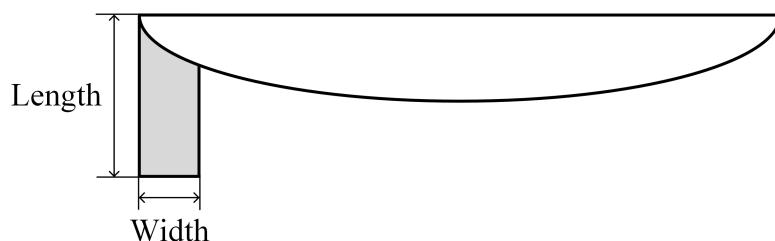


Figure 4.12: Rudder dimensions

The calculation for the relative water velocity on the rudder is the same as the case for a keel. However the angle of attack slightly differs. Assume that $\beta_{vk} = \beta_{vr}$, the rudder angle of attack is therefor,

$$\alpha_r = -\beta_{vr} + \delta_r + \pi \quad (4.67)$$

where δ_r is the rudder angle. The rudder forces and moments can be calculated in the same manner as the keel,

$$R = \begin{bmatrix} F_{RL} \sin \beta_{vk} + F_{RD} \cos \beta_{vk} \\ (-F_{RL} \cos \beta_{vk} + F_{RD} \sin \beta_{vk}) \cos(\phi) \\ h_3 F_{RL} \cos \beta_{vk} - F_{RD} \sin \beta_{vk} \\ (l_4 F_{RL} \cos \beta_{vk} - F_{RD} \sin \beta_{vk}) \cos(\phi) \end{bmatrix} \quad (4.68)$$

where F_{RL} is the rudder lift force and F_{RD} is the rudder drag force.

4.7.3. Modeling of a Sail

The sail just as the rudder and keel is assumed to be a rigid foil. The sail of the sailboat is the only form of propulsion the sailboat has. The sail behaves similarly to a airplanes wing. The sail generates a force due to a pressure difference acting over the area. The drag force is always in the direction of apparent wind and the lift is perpendicular to the apparent wind. The sail of the Dragonflite 95 sailboat is divided in two sails, the main sail and the jib sail. It is assumed that the overall sail area is a combination of the two sails. Both sails are actuated through a single servo motor controlling the both sail angle.

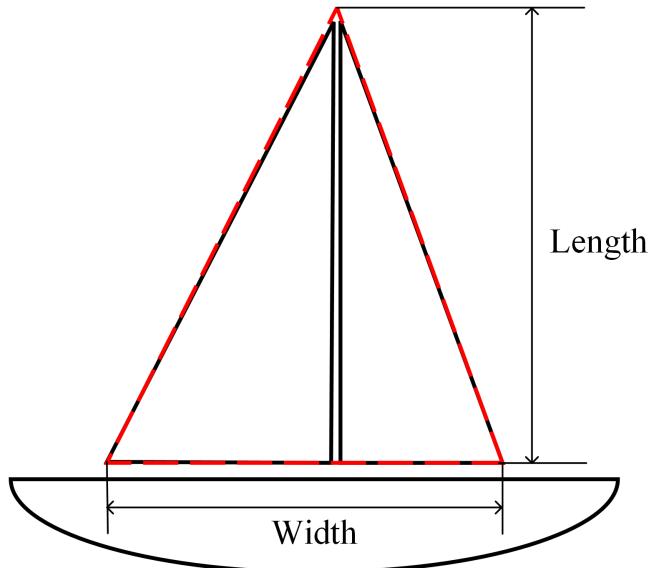


Figure 4.13: Sail Dimensions

The lift and drag coefficients are calculated in the same manner as in the case of a keel

and rudder. The apparent wind velocity is calculated in Equations 4.43, 4.45 and 4.46. The apparent wind angle γ_ω is already defined in Equation 4.41. The angle of attack is therefor,

$$\alpha_s = \gamma_\omega - \delta_s + \pi \quad (4.69)$$

The sail force is then formulated as,

$$S = \begin{bmatrix} -F_{SL} \sin(\gamma_\omega) + F_{SD} \cos(\gamma_\omega) \\ F_{SL} \cos(\gamma_\omega) + F_{SD} \sin(\gamma_\omega) \\ h_1(F_{SL} \cos(\gamma_\omega) + F_{SD} \sin(\gamma_\omega)) \\ F_{S_x} l_2 \sin(\gamma_\omega) - F_{S_y} (l_1 + l_2 \cos(\gamma_\omega)) \end{bmatrix} \quad (4.70)$$

where S_x and S_y is the sail force in the x and y body-frame axis. The sail force does not take into account when the sailboat has a roll angle significantly larger than zero. Now taking into account the roll angle the apparent wind angle is,

$$\gamma_{\omega\phi} = \tan^{-1}(\tan(\gamma_\omega) \cos(\phi)) \quad (4.71)$$

The apparent wind velocity when a non zero roll angle is present is,

$$V_{\gamma_{\omega\phi}} = V_{\gamma_\omega} \sqrt{1 - (\sin(\gamma_\omega) \sin \phi)^2} \quad (4.72)$$

From Equation 4.71 and 4.72 the lift force of the sail will decrease by the reduction of both the apparent wind angle and the apparent wind velocity. The decreasing ratio of force caused by the roll angle ϕ can be described as,

$$\frac{\gamma_{\omega\phi}}{\gamma_\omega} \frac{V_{\gamma_{\omega\phi}}}{V_{\gamma_\omega}} = \frac{\tan^{-1}(\tan(\gamma_\omega) \cos(\phi))}{\gamma_\omega} \left(1 - (\sin(\gamma_\omega) \sin \phi)^2\right) \quad (4.73)$$

In [25] it is shown how this term can be further analyzed by converting the force vector to the horizontal component of force and how the power series $\cos(\phi)^2$ approximates the total effect of roll angle on the sail forces and moments. The new forces and moments adapting for roll angle can be describe as,

$$S_n = \begin{bmatrix} S_X \cos(\phi)^2 \\ S_Y \cos(\phi)^2 \\ S_K / \cos(\phi) \\ S_N \cos(\phi)^2 \end{bmatrix} \quad (4.74)$$

4.8. Sailboat Differential Equation

The vector \mathbf{n} for a sailboat is described by the following set of differential equations:

$$\dot{x} = u \cos(\Psi) - v \sin(\Psi) \cos(\Phi) \quad (4.75)$$

$$\dot{y} = u \sin(\Psi) + v \cos(\Psi) \cos(\Phi) \quad (4.76)$$

$$\dot{\Psi} = r \cos(\Phi) \quad (4.77)$$

$$\dot{\Phi} = p \quad (4.78)$$

The vector \mathbf{v} for a sailboat is describe by the following set of differential equations:

$$\dot{u} = \frac{f_s \sin(\delta_s) - f_r \sin(\delta_r) - p_1 u}{m - X_{\dot{u}}} \quad (4.79)$$

$$\dot{v} = \frac{(-f_s \cos(\delta_s) + f_r \cos(\delta_r)) \cos(\Phi) - p_2 v}{m - Y_{\dot{v}}} \quad (4.80)$$

$$\dot{r} = \frac{(p_6 - p_7 \cos(\delta_s)) f_s - p_8 \cos(\delta_r) f_r - p_3 r}{I_z - N_{\dot{r}}} \quad (4.81)$$

$$\dot{p} = \frac{z_s f_s \cos(\delta_s) \cos(\Phi) - p_9 g \sin(\Phi) - p_{10} p}{I_x - K_{\dot{p}}} \quad (4.82)$$

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 4 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. The methods of control where chosen due to reliability in the case of model deviations as well as stability in the presences of disturbances. The combination of control systems are group under what is known as the motion control system.

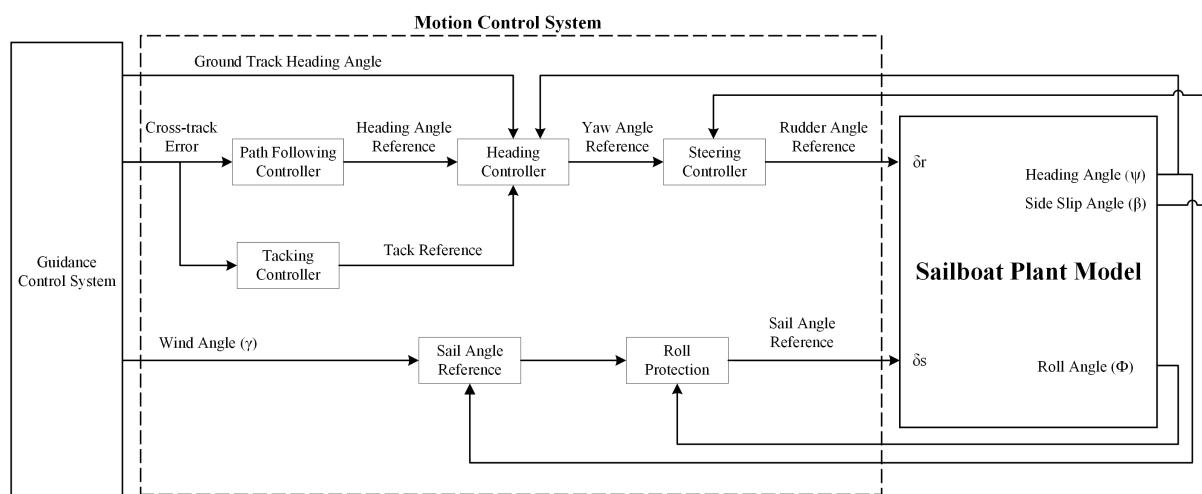


Figure 5.1: Block diagram overview of Sailing Control System

5.1.1. Motion Control System Overview

The motion control system is divided into its individual controllers and protectors. The controllers consist of:

- The steering controller which controls the rudder angle and is responsible for steering the sailboat in the direction of a specific yaw angle. The rudder angle is controlled through a servo motor.
- The heading controller is responsible for determining the appropriate yaw angle given the heading angles. The heading controller is also responsible for deciding when the sailboat should perform a tacking maneuver.
- The path following controller generates a heading angle according to a guidance system that aims to keep the sailboat on track. The track is the line generated between two waypoints.
- The tacking controller controls the tack reference for when the sailboat is required to tack. The controller makes use of the cross track error and wind angle to determine what tack should be performed. There are also more than one method of tacking. The controller is also capable of switching the type of tacking maneuver, which dictates how a sailboat will sail up wind.
- The sail angle reference block is not a controller but rather a calculation that calculates the optimal sail angle given the apparent wind angle.
- Roll protection is a novel control method that functions as a safety system that winches out the sail to decrease the sailboat's roll angle.

5.2. Steering Controller Design

To design the steering controller it is important to consider what is known as the stability and maneuverability of an ocean vessel. In this section the stability and maneuverability definitions are discussed and how traditional PID controller solve the issue of stabilizing an ocean vessel. The steering controller for the dragonflite 95 sailboat is determined through Nomoto's first order model and is easily calculated through practical test. The steering controller is then tested on the DP model in simulation with and without external forces to establish its performance.

5.2.1. First Order Nomoto Model

The Nomoto first order model [27] is a very common practice in designing a steering controller. The time domain representation of Nomoto's first order model is presented

below, where K is the gain constant and T is the time constant of the ocean vessel, assuming the ocean vessel is traveling at a constant speed u_0 .

$$T\dot{r} + r = K\delta \quad (5.1)$$

with the notation

$$\dot{\psi} = r \quad (5.2)$$

where ψ is the heading of the ship. Equation 5.1 can be written as

$$T\ddot{\psi} + \dot{\psi} = K\delta \quad (5.3)$$

where

$$x = \begin{bmatrix} \psi \\ r \end{bmatrix} \quad (5.4)$$

$$u = \delta \quad (5.5)$$

$$y = \psi \quad (5.6)$$

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 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 ill-conditioning problem associated with the second order Nomoto model in the identification of the model parameters from input-output data outweighs the improvement gained in its modeling capability. The transfer function representation of Nomoto's first order model is shown below,

$$\frac{r(s)}{\delta(s)} = \frac{K}{Ts + 1} \quad (5.7)$$

Equation 5.7 can be integrated to give the relationship between rudder angle and yaw angle.

$$\frac{\psi(s)}{\delta(s)} = \frac{K}{Ts^2 + s} \quad (5.8)$$

This system has two poles one located at,

$$p_1 = 0; \quad (5.9) \qquad p_2 = -\frac{1}{T} \quad (5.10)$$

The second pole will always lie to the left of the imaginary axis due to T always being positive. The poles are illustrated in Figure 5.2.

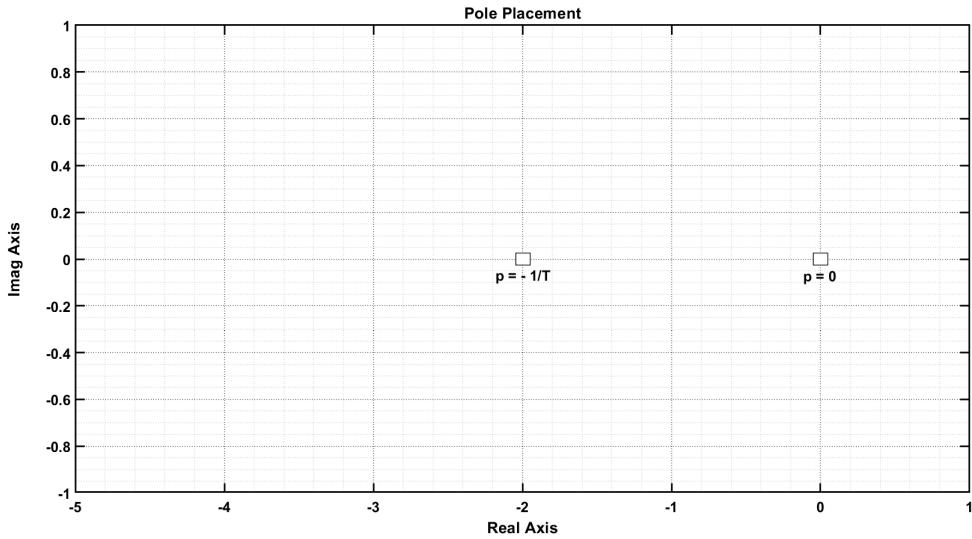


Figure 5.2: Pole placement

Due to the inability to control the speed of the sailboat u_0 , the values for K and T will be calculated at different speeds and then linear gain scheduling will be used to appropriately compensate for the sailboat changing dynamics. The values of K and T can be calculated using Kempf's Zig-Zag maneuver [28]. The calculation method is explained in detail in Section . The ratio of K and T is what is known as the maneuverability index of a ship P ,

$$P = \frac{K}{2T} \quad (5.11)$$

The maneuverability of a vessel is an indication how well a ship can maneuver itself. Generally for large ships a value of > 0.3 is good maneuverability. There is however a trade off between maneuverability and stability of a vessel. The stability and maneuverability will now be discussed and how it leads to the need for a PID controller in the steering of a vessel.

5.2.2. Stability and Maneuverability

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 ocean vessel 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:

$$\dot{x} = u \cos(\psi) - v \sin(\psi) \approx u_0 \cos(\psi) \quad (5.12)$$

$$\dot{y} = u \sin(\psi) + v \cos(\psi) \approx u_0 \sin(\psi) \quad (5.13)$$

$$\dot{\psi} = r \quad (5.14)$$

$$T\ddot{r} + r = K\delta + \omega \quad (5.15)$$

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:

$$\delta = -K_p(\psi - \psi_d) - K_d r \quad (5.16)$$

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 5.16, into Nomoto's first-order model, in Equation 5.14, yields the closed-loop system

$$T\ddot{\psi} + (1 + KK_d)\dot{\psi} + KK_p\psi = KK_p\psi_d + \omega \quad (5.17)$$

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

$$m\ddot{\psi} + d\dot{\psi} + k\psi = f(t) \quad (5.18)$$

with driving input

$$f(t) = k\psi_d + \omega \quad (5.19)$$

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

$$\lambda_{1,2} = \frac{-d \pm \sqrt{d^2 - 4km}}{2m} \quad (5.20)$$

$$\omega_n = \sqrt{\frac{k}{m}} \quad (5.21)$$

$$\xi = \frac{d}{2} \frac{1}{\sqrt{km}} \quad (5.22)$$

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

$$\lambda_1 = -\frac{d}{m} = -\frac{1}{T} > 0 \quad (5.23)$$

$$\lambda_2 = 0 \quad (5.24)$$

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

$$\lambda_1 = -\frac{d}{m} = -\frac{1}{T} < 0 \quad (5.25)$$

$$\lambda_2 = 0 \quad (5.26)$$

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

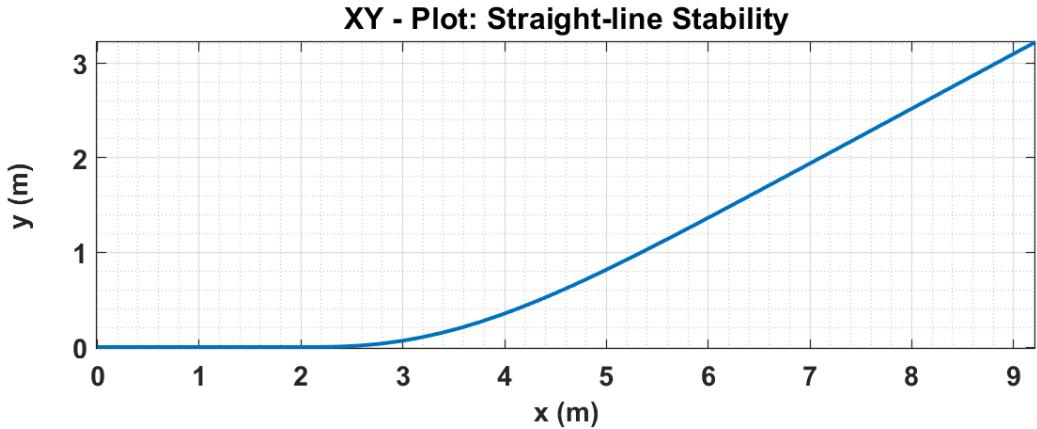


Figure 5.3: XY - plot of straight-line stability

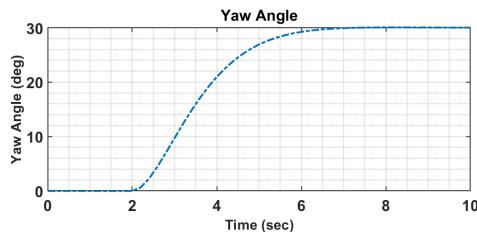


Figure 5.4: Yaw

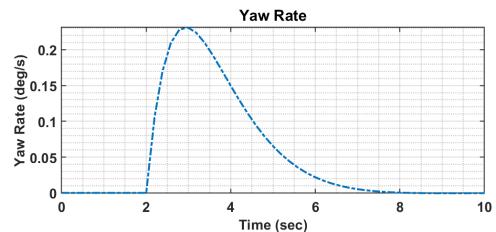


Figure 5.5: Yaw Rate

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

$$\operatorname{Re}\{\lambda_{1,2}\} < 0 \quad (5.27)$$

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

$$\lambda_{1,2} = \frac{-d \pm \sqrt{d^2 - 4km}}{2m} = \left(-\xi \pm \sqrt{\xi^2 - 1} \right) \omega_n < 0 \quad (5.28)$$

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

$$\lambda_{1,2} = \frac{-d \pm j\sqrt{d^2 - 4km}}{2m} = \left(-\xi \pm j\sqrt{\xi^2 - 1} \right) \omega_n \quad (5.29)$$

Directional stability for a critically damped and under damped ocean vessel is illustrated

in Figure 5.6 and 5.9. 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.

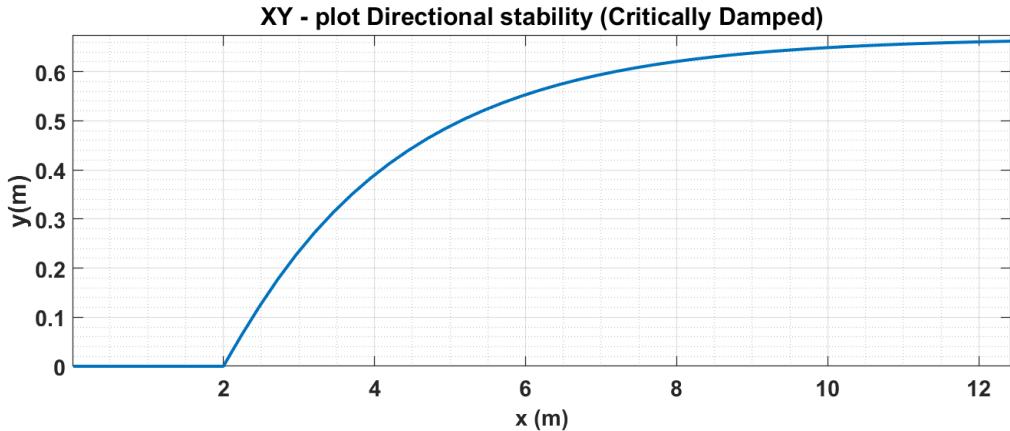


Figure 5.6: XY - plot of directional stability (critically damped)

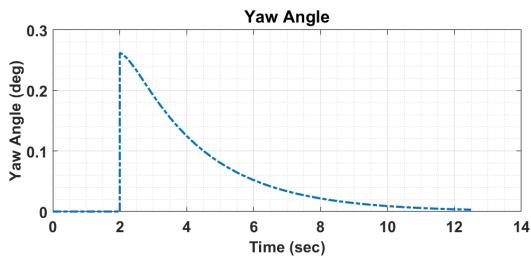


Figure 5.7: Yaw

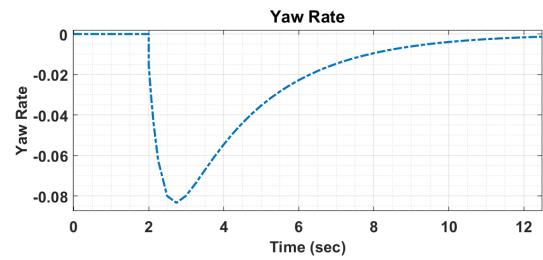


Figure 5.8: Yaw Rate

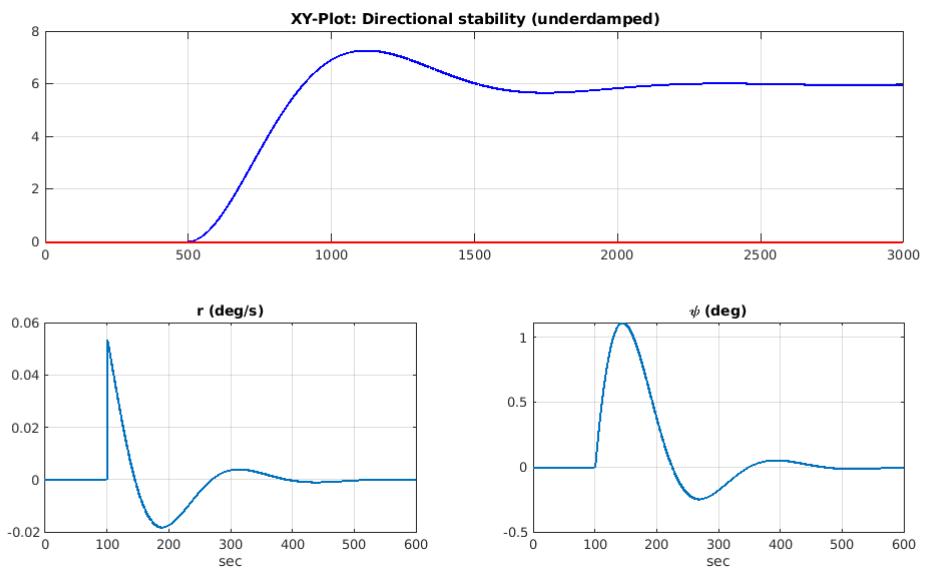


Figure 5.9: Directional Stability(under damped)

Positional Motion Stability: Positional motion stability implies that the ship should return to its original path after a disturbance, illustrated in Figure 5.10. 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.

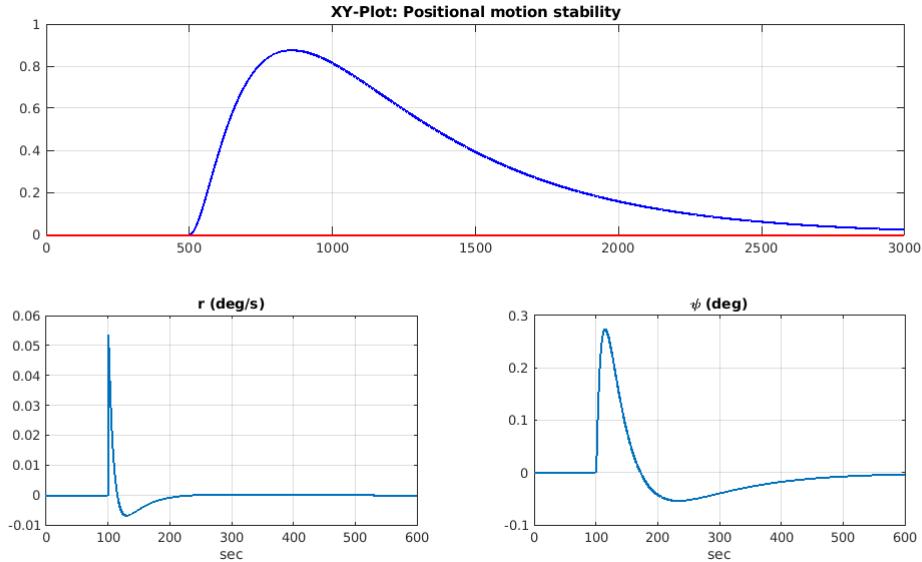


Figure 5.10: Positional Motion Stability

The proposed PID controller is illustrated in Figure 5.11. 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.

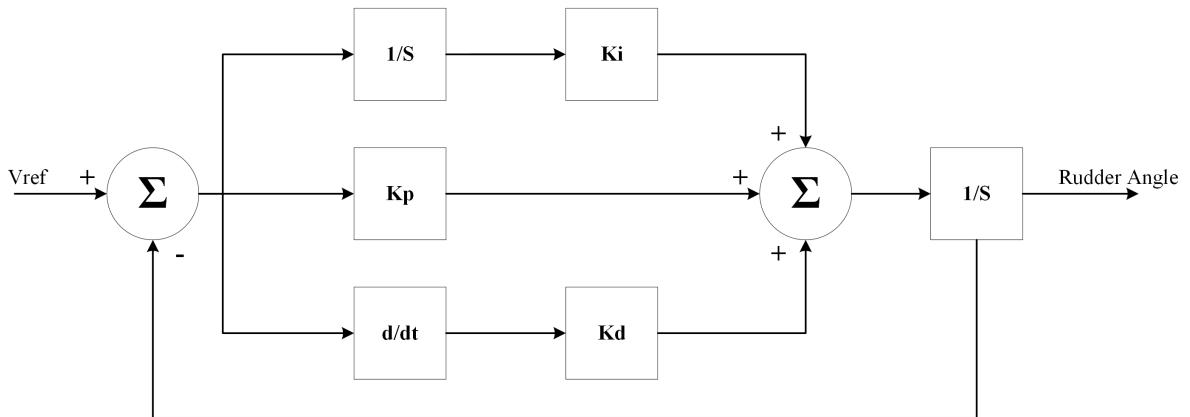


Figure 5.11: PID rudder controller

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

$$\delta_r = K_p e + K_I \int_0^t e(\tau) d\tau + K_D \frac{de}{dx} \quad (5.30)$$

where,

$$e = \psi_d - \psi \quad (5.31)$$

Now that it is established why a PID controller is required to fully stabilize an ocean vessel the values for the K_i , K_i and K_D needs to be calculated. The controller gains will be designed via the root locus method. The calculated transfer function is illustrated below,

$$\frac{\psi}{\delta_r} = \frac{0.6252}{0.4s^2 + s} \quad (5.32)$$

The desired time specifications are the settling time $T_s = 2$ seconds with an overshoot of 0.1. The controller has the following transfer function,

$$D(s) = \frac{K(s+a)(s+b)}{s} \quad (5.33)$$

The damping coefficient and natural frequency for the desired poles is,

$$\zeta = 0.59121 \quad (5.34) \qquad \omega_n = 3.383 \quad (5.35)$$

The desired poles are located at,

$$p_1 = -2 + j2.7288 \quad (5.36) \qquad p_2 = -2 - j2.7288 \quad (5.37)$$

Firstly the PD part of the controller, $K(s+a)$, will be designed through the root locus method. The open-loop root locus are illustrated below,

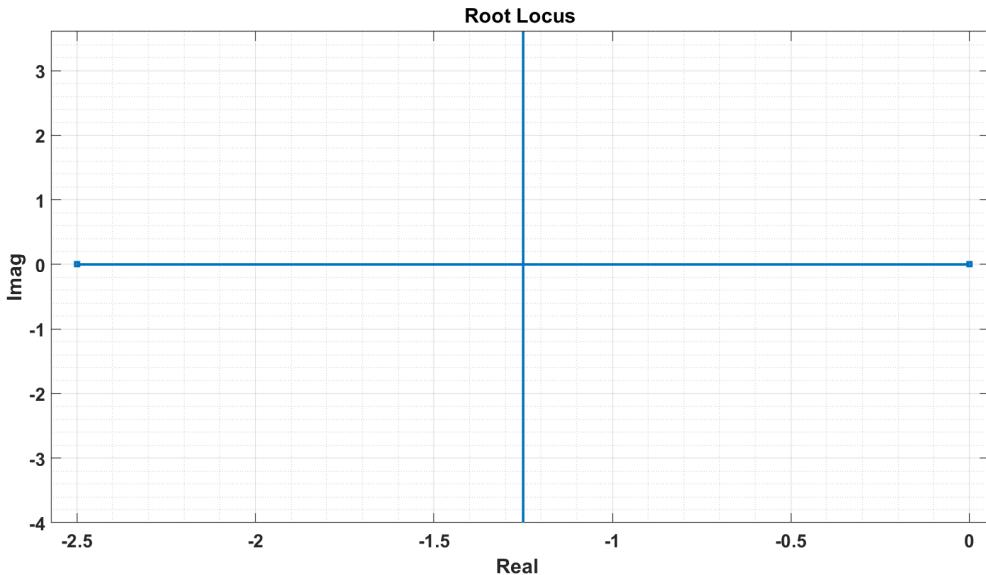


Figure 5.12: Root Locus of system

The zero's position a will be placed using the departure angle rule of poles and zeros. The value of a is calculated as $a = 7.6307$. The new root locus are illustrated in Figure 5.13

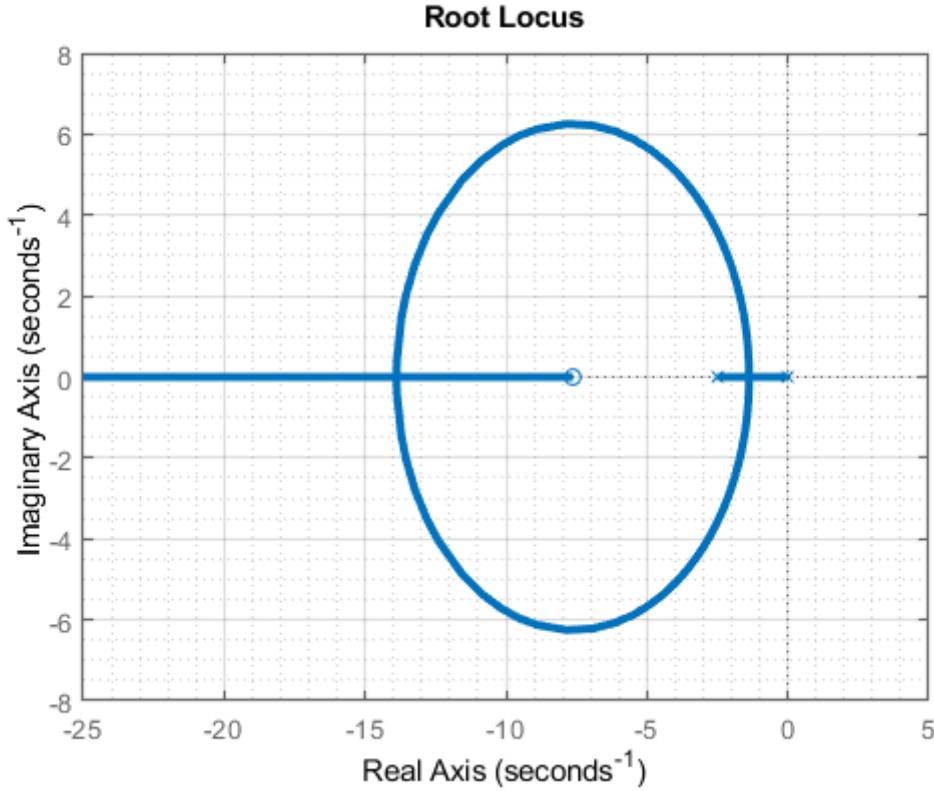


Figure 5.13: Root Locus of plant with PD Controller

The gain of the PD Controller is designed using the law of magnitude condition. The gain is therefore equal to $K = 0.96$. The PD Controller is shown below,

$$D_{PD}(s) = 0.96(s + 7.6307) \quad (5.38)$$

The integrator is placed at the origin and another zero is placed close to the origin, $b = 0.01$, to cancel out the integrator so that there is minimal affect on the root locus. The full PID controller is thus,

$$D_{PID}(s) = \frac{0.96(s + 7.6307)(s + 0.01)}{s} \quad (5.39)$$

The new augmented system is equal to

$$D_{PID}(s)H(s) = \frac{0.6(s + 7.6307)(s + 0.01)}{s(0.4s^2 + s)} \quad (5.40)$$

The root Locus is illustrated in Figure 5.14.

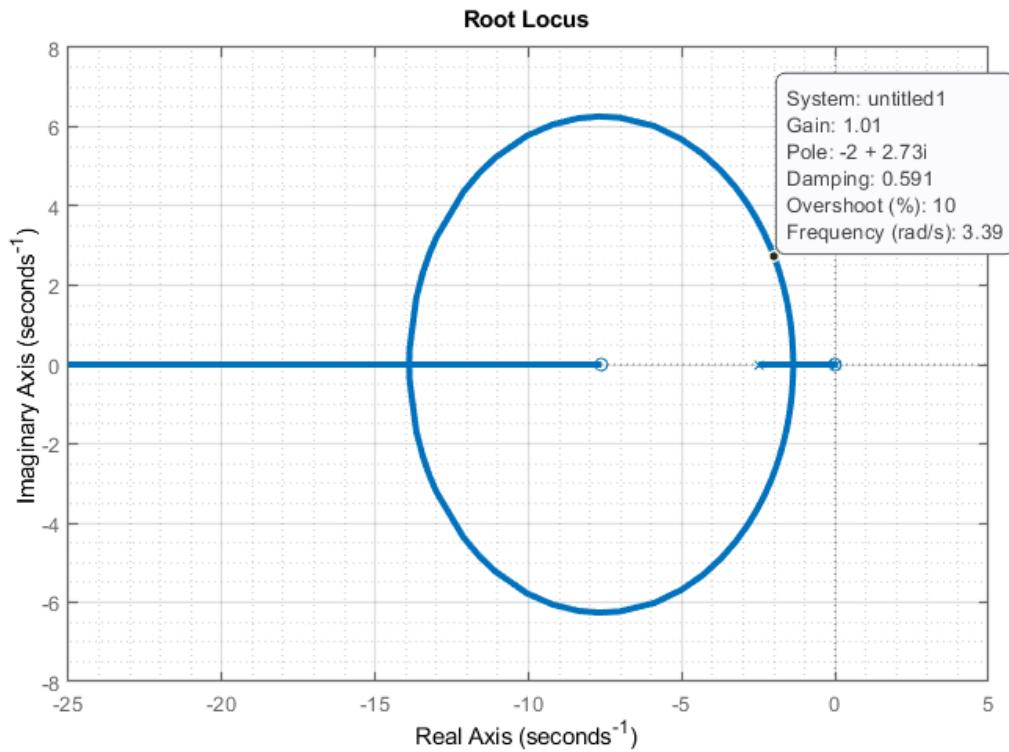


Figure 5.14: Augmented system root Locus

The individual gains are $K_P = 7.333$, $K_D = 0.96$ and $K_I = 0.07$. The step response is illustrated in Figure 5.15.

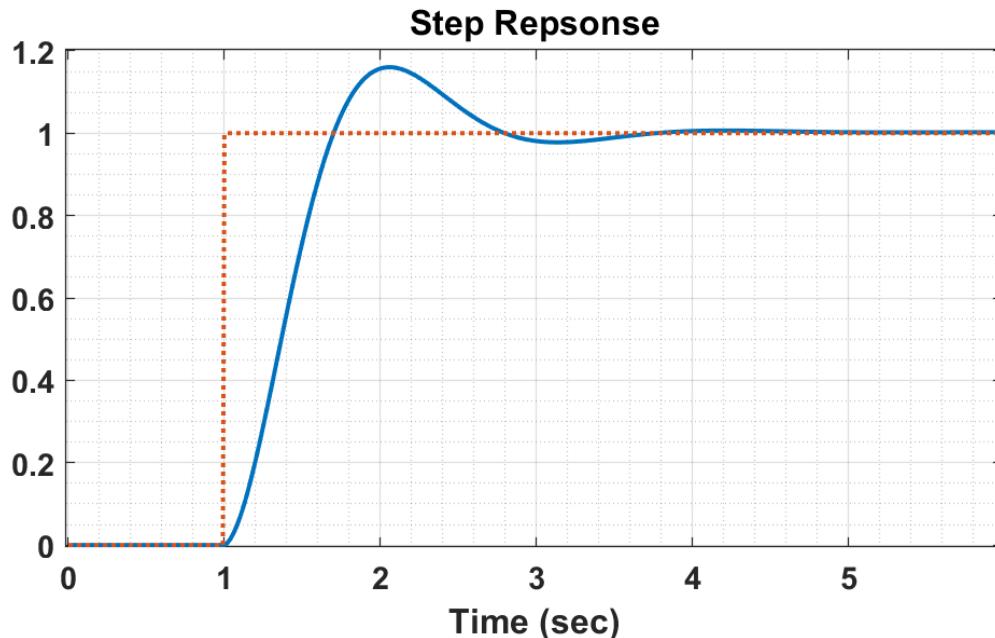


Figure 5.15: Step response of system

5.2.3. Acceleration Feedback

It is possible to extend the results obtained for the use of a PID to include acceleration feedback. As shown in Equation 5.18, that the steering dynamics can be presented as a mass-damper system. The system is expanded to include acceleration feedback shown below,

$$m\ddot{\psi} + d\dot{\psi} + k\psi = \tau_{PID} - K_m\ddot{\psi} + \omega \quad (5.41)$$

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

$$(m + K_m)\ddot{\psi} + d\dot{\psi} + k\psi = \tau_{PID} + \omega \quad (5.42)$$

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,

$$\tau = \tau_{PID} - h_m(s)\ddot{x} \quad (5.43)$$

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

$$h_m(s) = \frac{K_m}{T_m s + 1} \quad (5.44)$$

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

$$\left(m + \frac{K_m}{T_m s + 1} \right) \ddot{\psi} + d\dot{\psi} + k\psi = \tau_{PID} + \omega \quad (5.45)$$

5.2.4. Proposed other Control Techniques

Another control law, proposed and tested in [2], for the rudder angle is using a non-linear proportional gain depending on the heading error shown below,

$$k(e) = \frac{k_p}{1 + c_p|e|} \quad (5.46)$$

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

$$\delta_r = \frac{k_p}{1 + c_p|e|} e + K_I \int_0^t e(\tau) d\tau + K_D \frac{de}{dx} \quad (5.47)$$

5.3. Sail Reference Angle

The force generated by the sail is shown in Equation 5.48 with $f_s = ||F_s||$.

$$F_s = \begin{bmatrix} f_s \sin(\delta_s) \\ -f_s \cos(\delta_s) \cos(\phi) \end{bmatrix} \quad (5.48)$$

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

$$AoA = \pi - (\delta_s - \psi_{aw}) \quad (5.49)$$

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

$$f_s = -K v_{aw}^2 \sin(\delta_s - \psi_{aw}) \sin(\delta_s) \quad (5.50)$$

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 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 [29], 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,

$$\delta_s(t) = \delta_s^b(t) + \alpha_u \quad (5.51)$$

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

$$\alpha_u = K_p e_u(t) + K_i \int_0^t e_u(\tau) d\tau \quad (5.52)$$

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

$$e_u(t) = u_{max} - u(t) \quad (5.53)$$

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 5.16 the optimal sail angle is plotted against the apparent wind angle.

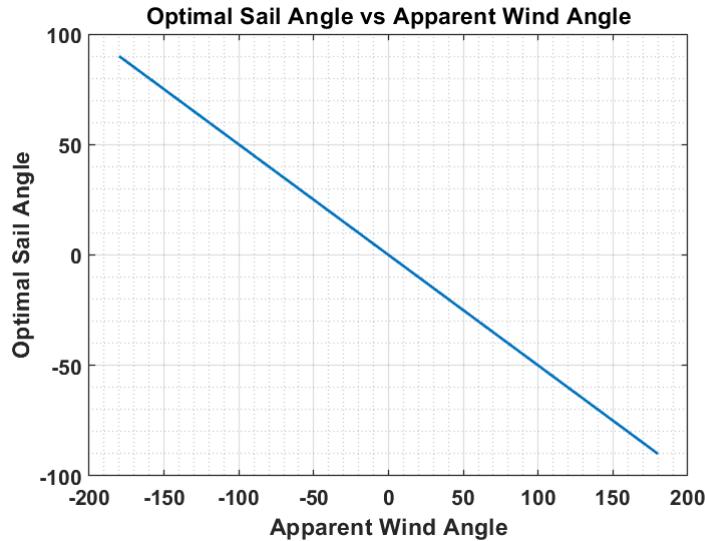


Figure 5.16: Optimal sail angle for varying wind angle

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

$$\delta_s = -0.5 \times \psi_{aw} \quad (5.54)$$

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 5.17, the linear approximation of the optimal sail produces a sail force greater or equal than zero in the whole range of apparent wind angle.

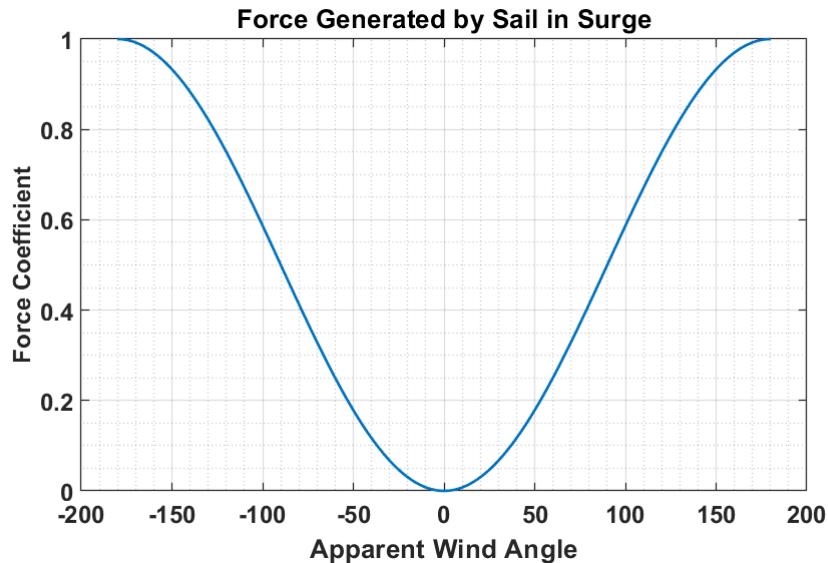


Figure 5.17: Dimensionless sail force

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

5.4. Roll Protection

The general saying that an upright ship is a happy ship applies to sailboats. An increased roll angle will increase the velocity in the surge direction as indicated in Equation 4.75. Therefore the roll angle should be kept at a minimum for the ship to sail straight. As already stated the roll rate is determined by the differential equation below,

$$\dot{p} = \frac{z_s f_s \cos(\delta_s) \cos(\phi) - p_9 g \sin(\phi) - p_{10} p}{I_x - K_p} \quad (5.55)$$

The three terms responsible contributing to the roll rate two of them are restoration forces acting on stabilizing the vessel and returning it to $\phi = 0$. The only controllable term is $z_s f_s \cos(\delta_s) \cos(\phi)$, where the controllable variable is the sail angle δ_s . The roll protection will be implemented by winching out the sail relative to a increase in roll angle. That means if the roll angle is positive the sail angle will be increased and when the roll angle is negative the sail angle will be decreased. The proposed increase/decrease in sail angle is described by the following equation,

$$\delta_p = a\phi \quad (5.56)$$

where a is used as multiplier that scales how strong the roll protection will be. Therefore the modified sail angle is,

$$\delta_s = \delta_{sref} + \delta_p \quad (5.57)$$

The roll protection will be tested on the actual sailboat to determine an appropriate roll multiplier value for a .

5.5. Tacking Controller

The tacking controller is used to calculate the required heading of a sailboat when the desired heading is in the no-go zone. The tacking maneuvers are divided into a direct tack and indirect tack. Direct tack is defined as the tack direction is dependent on the cross-track error of the sailboat, whereas an indirect tack is when the sailboat aims to only tack twice to reached the desired waypoint. Direct tacking is favoured in high accuracy moments, but requires a higher velocity for the sailboat to successfully perform multiple tacks. Indirect tacking however is more susceptible to disturbances such as drift but will let the sailboat reaches its destination quicker. The state machine logic, illustrated in Figure 5.18, is used for both forms of tacking, but the decisions on when to tack changes in different tacking methods.

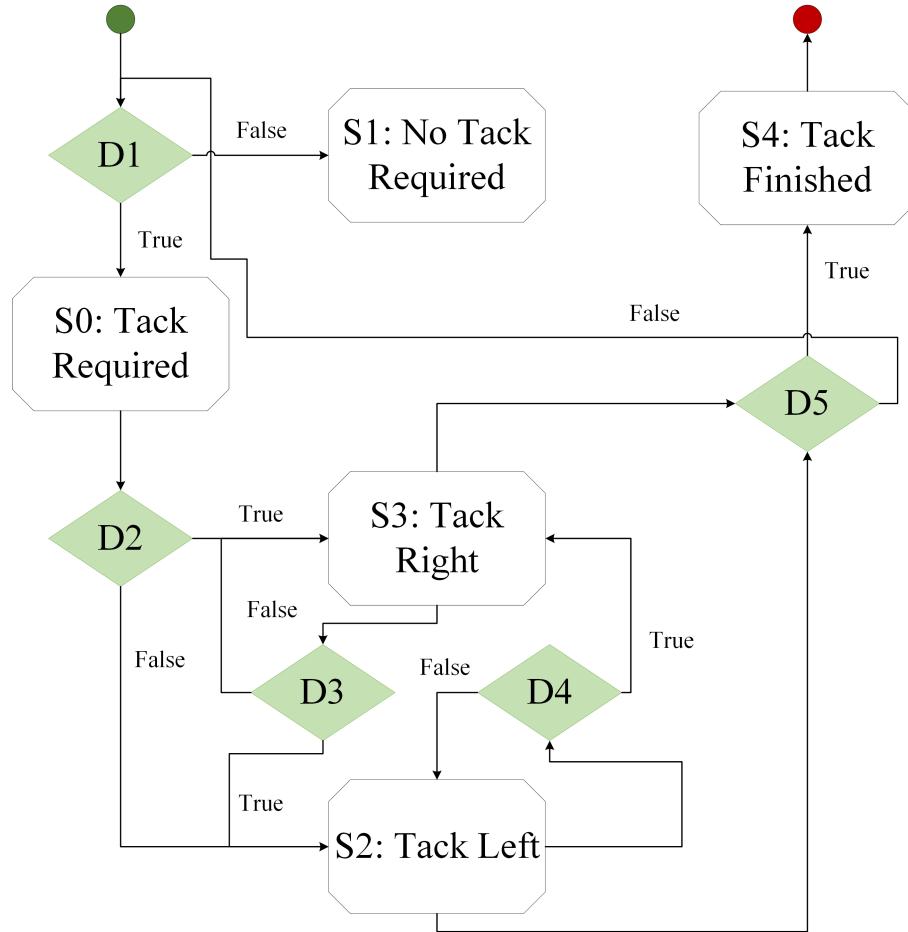


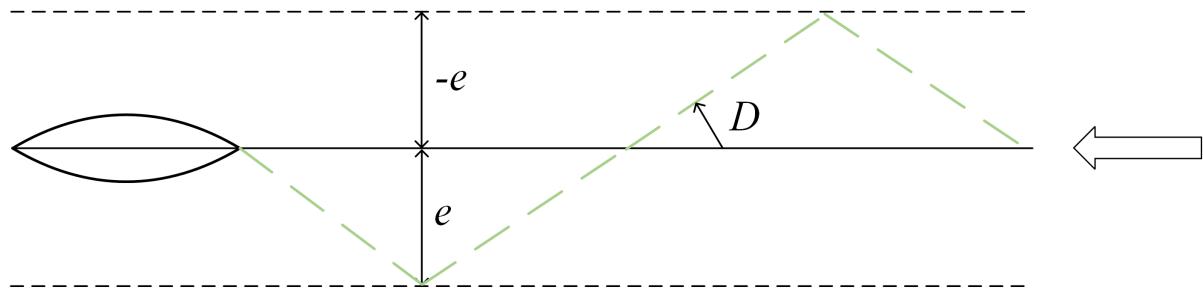
Figure 5.18: State Diagram for tacking controller

Direct Tack

The heading controller for a direct tack maneuver is a fuzzy logic controller that has the following inputs, tack direction, tack distance, tack left angle and tack right angle. The cross-track error is defined in Section 6.1. The direct tack maneuver is illustrated in Figure 5.19. The arrow is indicating the apparent wind direction, e is the cross-track error and D is the tack angle. The parameter D and e is tunable to achieve the best result. The direct tack decisions are shown in Table 5.1.

Table 5.1: Direct tacking controller state machine decisions

Decision	Definition
D1	Heading angle is in no-go zone? ($ \text{apparent wind angle} < 45^\circ$)
D2	Is the heading angle closest to a right tack?
D3	Is the cross-track error larger than specified?
D4	Is the cross-track error smaller than specified?
D5	Is the waypoint reached?

**Figure 5.19:** Direct tack maneuver

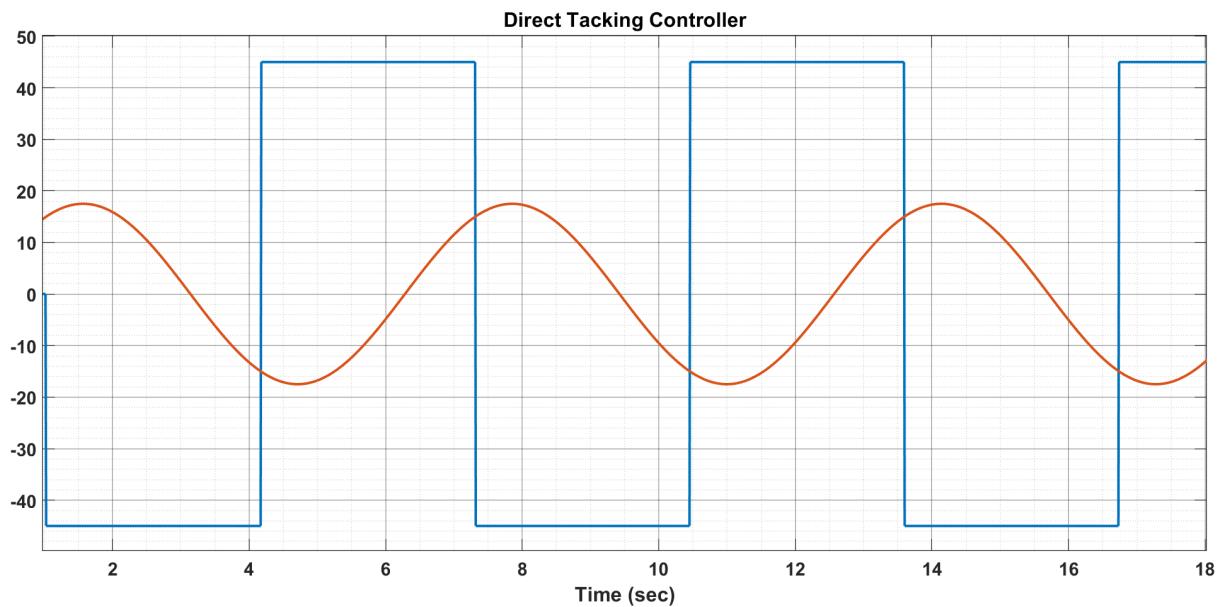
The algorithm for the fuzzy logic direct tacking controller is illustrated in Algorithm 5.1. The output of the controller is illustrated in Figure 5.20. In that exampled $e = 15$, $D = 45$ and $\gamma = 0$.

Algorithm 5.1: Fuzzy logic Direct tacking controller

Require: State 1: Direct Tacking

```

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

**Figure 5.20:** Output and input of fuzzy logic controller

Indirect Tack

An indirect tack as mentioned is when the sailboat aims to tack only twice to reach the destination. The difference between the direct and indirect tack is when the sailboat

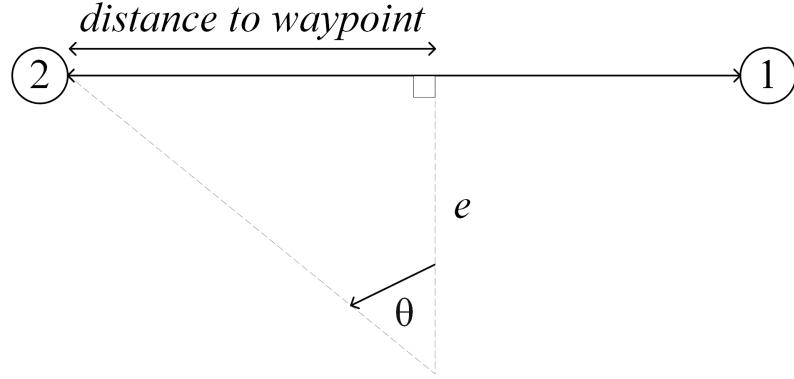


Figure 5.21: Indirect tack calculation

decides to tack. The decision to switch tacks is calculated when the angle θ is equal to 45° as indicated in Figure 5.21. The angle θ is calculated as,

$$\theta = \tan^{-1} \left(\frac{e}{\text{distance to waypoint}} \right) \quad (5.58)$$

When this angle is equal to the tack angle the sailboat will perform a tack. The decisions for indirect tacking is shown in Table 5.2. The algorithm that determines when the sailboat needs to switch tacks are indicated in Algorithm 5.2.

Table 5.2: Indirect tacking controller state machine decisions

Decision	Definition
D1	Heading angle is in no-go zone? ($ \text{apparent wind angle} < 45^\circ$)
D2	Is the heading angle closest to a right tack?
D3	Is the desired heading angle equal to a left tack angle?
D4	Is the desired heading angle equal to a right tack angle?
D5	Is the waypoint reached?

Algorithm 5.2: Fuzzy logic indirect tacking controller

Require: State 2: Indirect Tacking

θ = angle to waypoint

γ_ω = apparent wind angle

ψ_K = tack angle + γ_ω

if $\psi_K < \theta$ **then**

State: Tack Left

else if $-\psi_K > \theta$ **then**

State: Tack Right

end if

Zone Tack

The third tacking controller is named the zone tack controller. The tacking controller now also takes into account wind shifts when tacking. The controller is the most complete controller and should give the best results in a environment where the wind changes continuously. Firstly the zones need to be defined,

5.6. Heading Controller

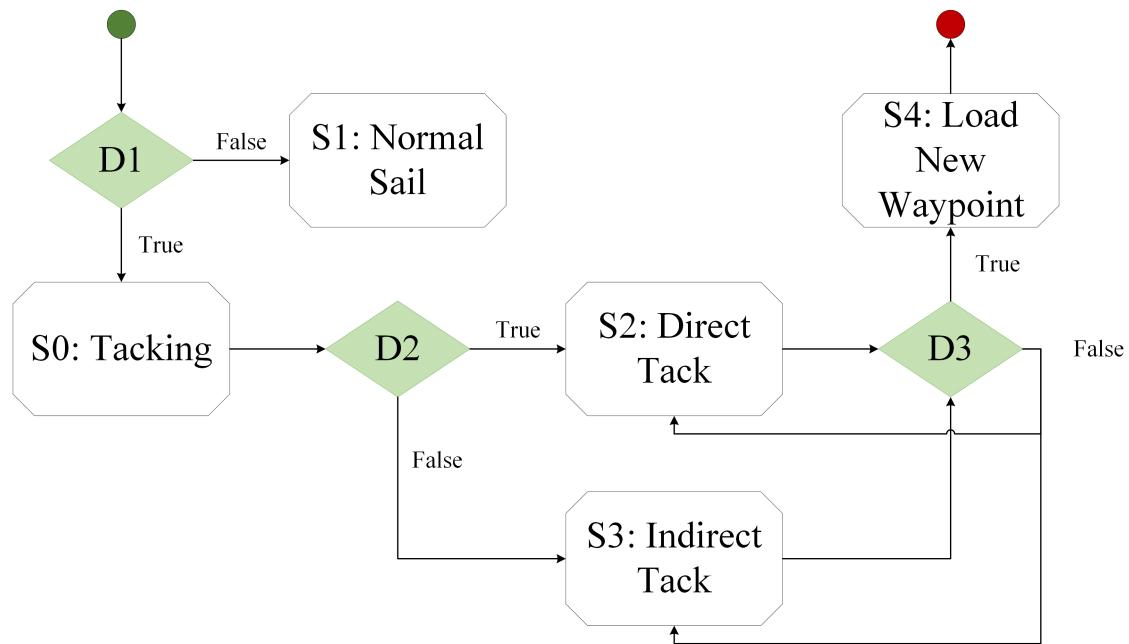


Figure 5.22: State diagram for heading controller

Chapter 6

Guidance Control System Development

6.1. Guidance System

The guidance system is similar to the aircraft guidance system, discussed in [30]. 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

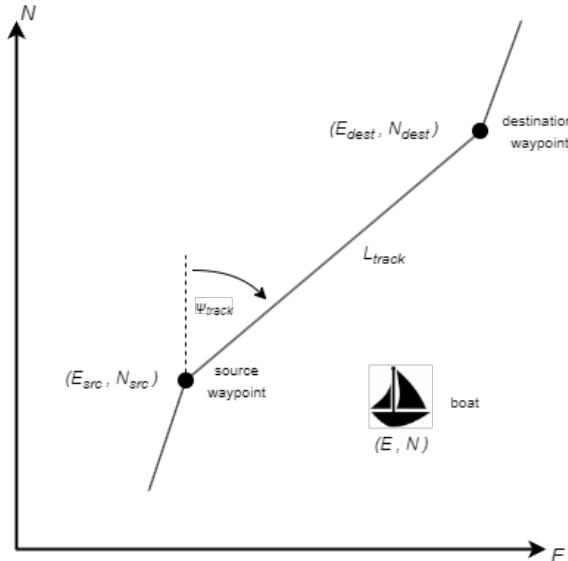


Figure 6.1: Ground track between source waypoint and destination waypoint

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

$$\psi_{track} = \tan^{-1} \left(\frac{E_{dest} - E_{src}}{N_{dest} - N_{src}} \right) \quad (6.1)$$

$$l_{track} = \sqrt{(N_{dest} - N_{src})^2 + (E_{dest} - E_{src})^2} \quad (6.2)$$

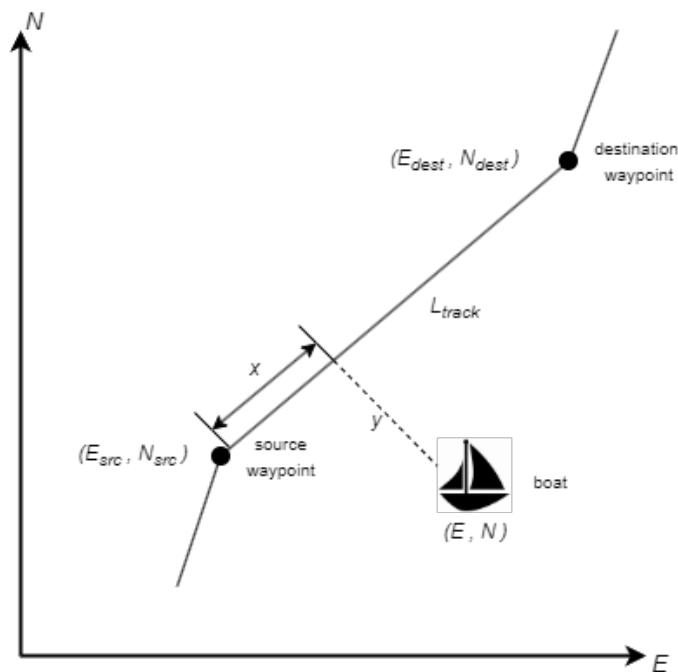


Figure 6.2: Cross-track error and in-track distance along track

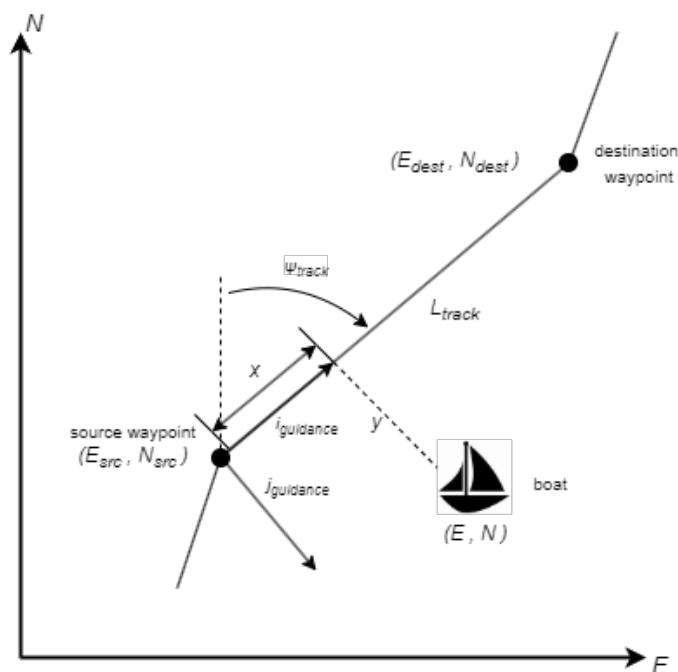


Figure 6.3: Guidance Axis System

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

$$\begin{bmatrix} s \\ e \end{bmatrix} = \begin{bmatrix} \cos(\psi_{track}) & \sin(\psi_{track}) \\ -\sin(\psi_{track}) & \cos(\psi_{track}) \end{bmatrix} \begin{bmatrix} N - N_{src} \\ E - E_{src} \end{bmatrix} \quad (6.3)$$

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.

6.2. Straight path Generation based on Waypoints

For surface craft vessels, discussed in [14], only two coordinates (x_k, y_k) for $k = 1, \dots, n$. The waypoint database therefore consists of

$$wpt.pos = (x_0, y_0), (x_1, y_1), \dots, (x_n, y_n) \quad (6.4)$$

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

$$wpt.speed = U_0, U_1, \dots, U_n \quad (6.5)$$

$$wpt.heading = \psi_0, \psi_1, \dots, \psi_n \quad (6.6)$$

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 [31] 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 6.4 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

$$wpt.radius = R_0, R_1, \dots, R_n \quad (6.7)$$

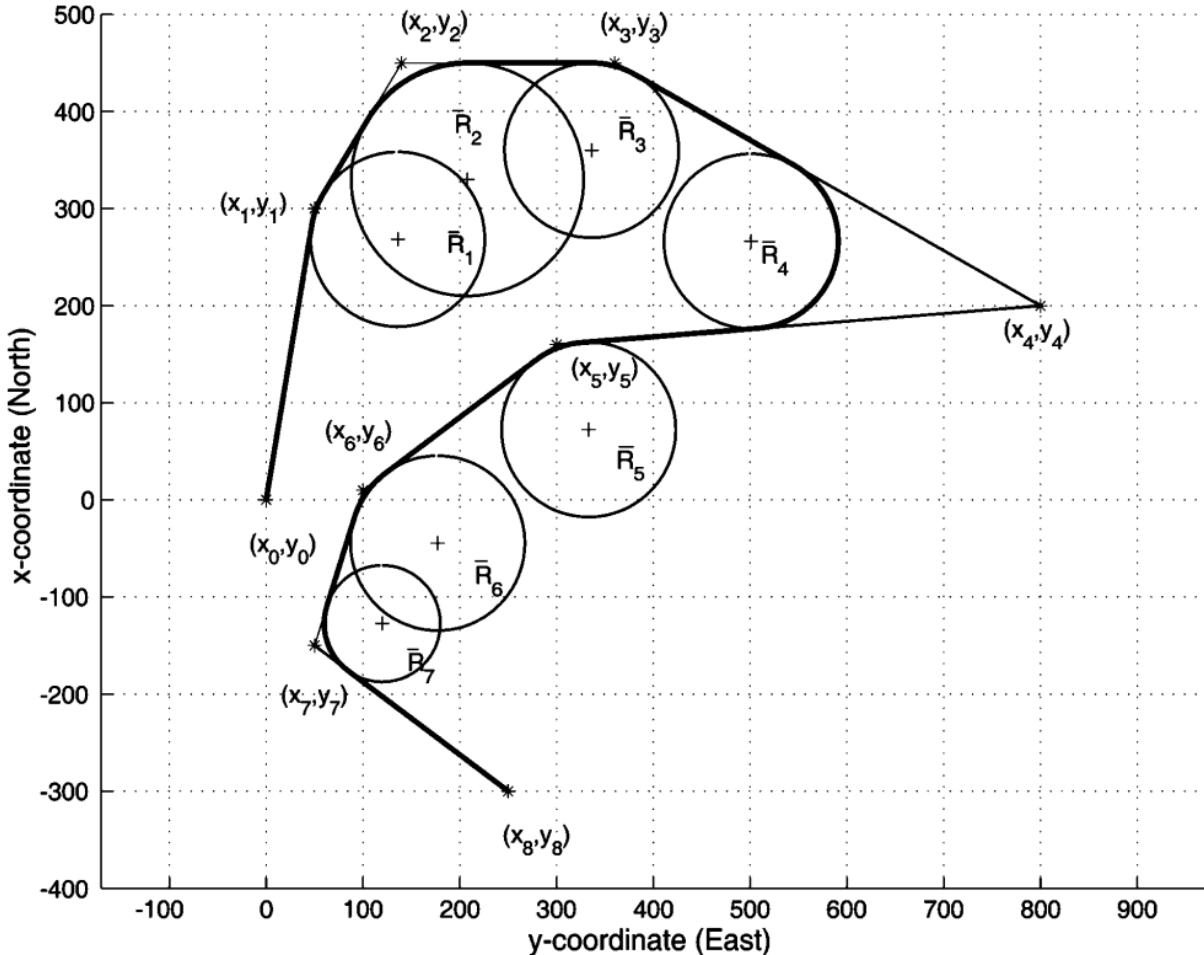


Figure 6.4: Straight lines and circular arc segments for waypoint guidance

6.3. Line-of-Sight Steering Law

In order to steer the boat on a path a method known as the *lookahead-based steering* [32]. 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

$$\lim_{t \rightarrow \infty} e(t) = 0 \quad (6.8)$$

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

$$x_d(e) = x_p + x_r(e) \quad (6.9)$$

where

$$x_p = \alpha_k \quad (6.10)$$

is the *path – tangential angle* illustrated in Figure 6.5, while

$$x_r(e) = \arctan\left(\frac{-e}{\Delta}\right) \quad (6.11)$$

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.

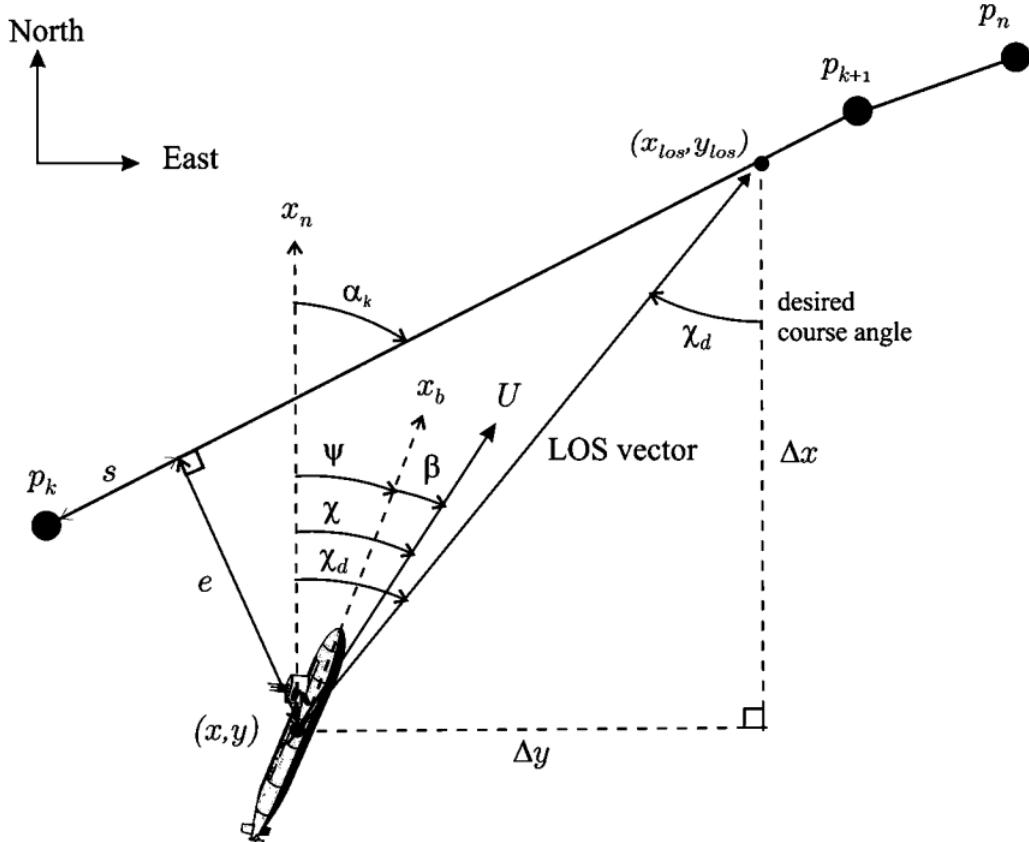


Figure 6.5: LOS guidance where the desired course angle x_d is chosen to point toward the LOS intersection point (x_{los}, y_{los})

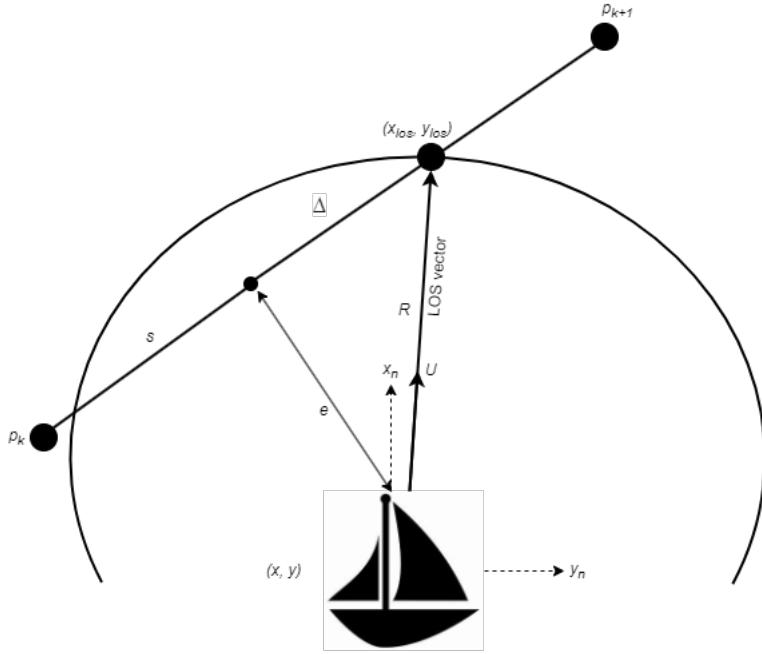


Figure 6.6: Circle of acceptance with constant radius R

Figure 6.6 illustrates that the circle of acceptance radius R is equal to

$$e(t)^2 + \Delta(t)^2 = R^2 \quad (6.12)$$

with

$$\Delta(t) = \sqrt{R^2 - e(t)^2} \quad (6.13)$$

varying between 0 and R for $|e(t)| = R$ and $|e(t)| = 0$, respectively. The steering law, in Equation 6.11, can also be interpreted as a saturating control law,

$$x_r(e) = \arctan(-K_p e) \quad (6.14)$$

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 6.6, 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,

$$x_r(e) = \arctan \left(-K_p e - K_i \int_0^t e(\tau) d\tau \right) \quad (6.15)$$

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

$$x_d(e) = \alpha_k + x_r(e) \quad (6.16)$$

with $x_r(e)$ as in Equation 6.14. 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.

6.4. 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,

$$x_d(e) = x_p + x_r(e) = \alpha_k + \arctan(-K_p e) \quad (6.17)$$

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 6.7,

$$\psi_d = x_d - \beta \quad (6.18)$$

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

$$\psi = \psi - \psi_d = \psi - x_d + \beta \quad (6.19)$$

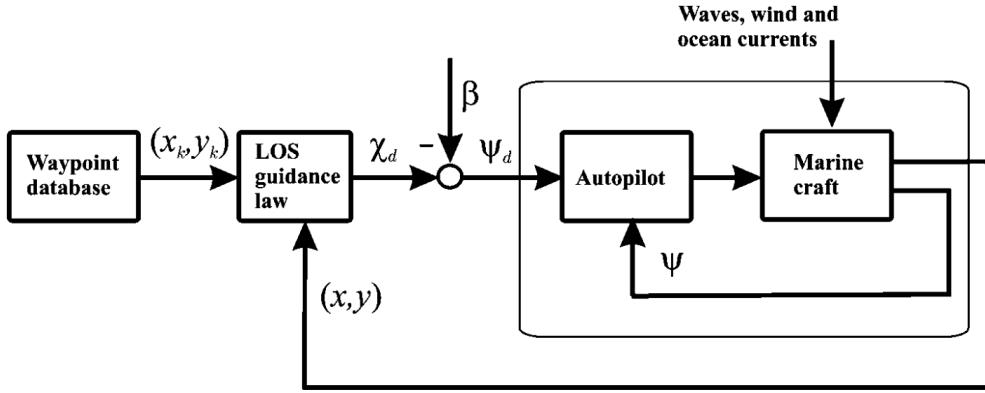


Figure 6.7: LOS guidance principle where the side slip angle B can be applied and compensated for by using integral action

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

$$\beta = \arcsin\left(\frac{v}{U}\right) \quad (6.20)$$

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,

$$\theta = \cos^{-1}\left(\frac{\bar{u} \cdot \bar{v}}{\|\bar{u}\| \|\bar{v}\|}\right) \quad (6.21)$$

The vectors \bar{u} and \bar{v} is illustrated in Figure 6.8. The vector \bar{u} is defined using the reference angle ψ_d and the vector \bar{v} is defined using the heading angle ψ .

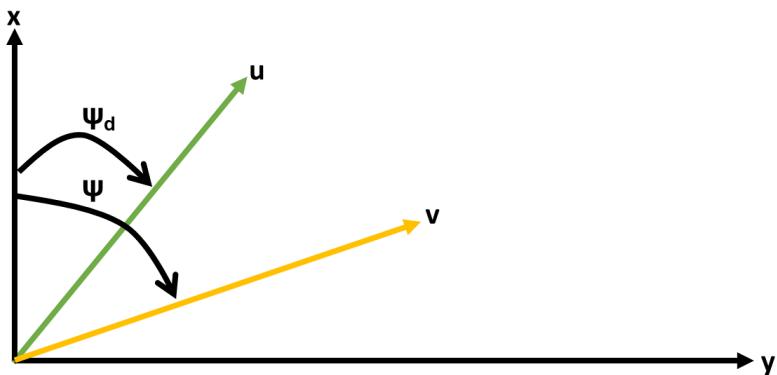


Figure 6.8: Definition of vector \bar{u} and \bar{v}

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

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos(\psi) \\ \sin(\psi) \end{bmatrix} \quad (6.22)$$

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

$$\bar{u} \cdot \bar{v} = x_{\bar{u}} x_{\bar{v}} + y_{\bar{u}} y_{\bar{v}} \quad (6.23)$$

$$\|\bar{u}\| = \sqrt{x_{\bar{u}}^2 + y_{\bar{u}}^2} \quad (6.24)$$

$$\|\bar{v}\| = \sqrt{x_{\bar{v}}^2 + y_{\bar{v}}^2} \quad (6.25)$$

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

$$\psi_d = -\theta \quad (6.26)$$

6.5. 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,

$$[x_{k+1} - x(t)]^2 + [y_{k+1} - y(t)]^2 \leq R_{k+1}^2 \quad (6.27)$$

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.

$$R = b \frac{y_{k+1} - y_k}{x_{k+1} - x_k} \quad (6.28)$$

where the value b is a constant value.

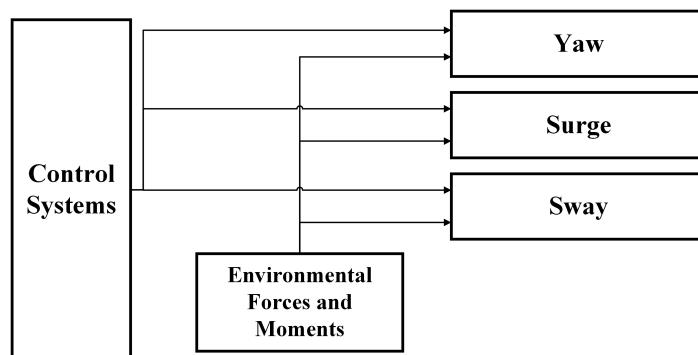
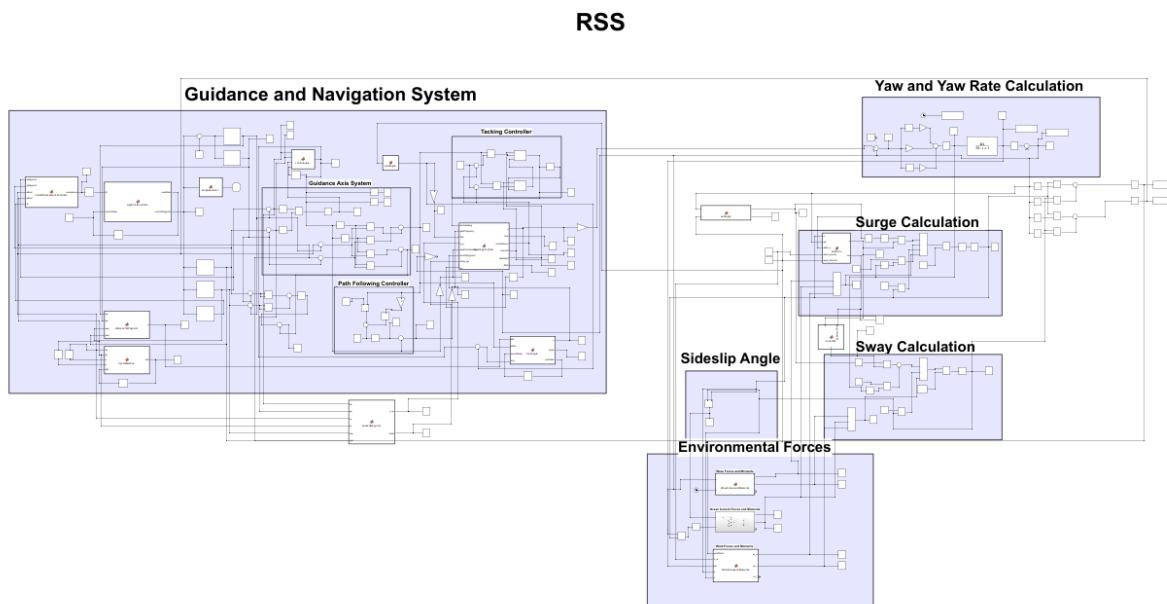
Chapter 7

Simulations

In this chapter the simulations and their results are discussed. The first simulation is done in Matlab where a non-linear 3-DOF simulation was developed. The simulation is modeled on the dragonflite 95 sailboat. In this simulation the developed motion control system and the guidance control system was implemented and tested. The second simulation is done on Ardupilot's Software-In-The-Loop (SITL) platform. This simulation will form a basis to compare the Ardupilot's control systems compare with the control systems developed in this project. That leads to the next Chapter where the additions to Ardupilot is discussed.

7.1. RSS Simulator

The simulation is based on the work done in [15]. The simulator aims to model a dragonflite 95 sailboat. Where the modelled parameters are partially obtained through numerical estimates and practical tests. The parameter estimations are discussed in Appendix A. The yaw and yaw rate differential equations are modeled through Nomoto's first order system. This equation is explained in depth in Section 5.2.1. The surge and sway differential equations are based on the equations mentioned in Section ... The 4 - DOF, roll, is not modeled because there is not a controller specifically designed to control the roll angle. An overview of the simulation is illustrated in Figure 7.2. The non-linear simulation forms a test environment where controllers and control systems is tested before they are practically implemented. The block diagram overview of the simulator is shown below.

**Figure 7.1:** Block diagram overview of RSS**Figure 7.2:** Overview of Simulink simulation

7.2. Motion Control System Simulation

7.2.1. Steering Controller Simulation

7.2.2. Sail Reference Angle Simulation

7.2.3. Tacking Controller Simulation

7.2.4. Heading Controller Simulation

7.3. Guidance System Simulation

7.3.1. Path Following Controller Simulation

7.3.2. Waypoint Navigation Simulation

7.3.3. Race Course Simulation

Chapter 8

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.

8.1. 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 8.1 the Ardupilot is responsible for three things, the vehicle specific code, the sensor libraries and the hardware abstraction layer(HAL).

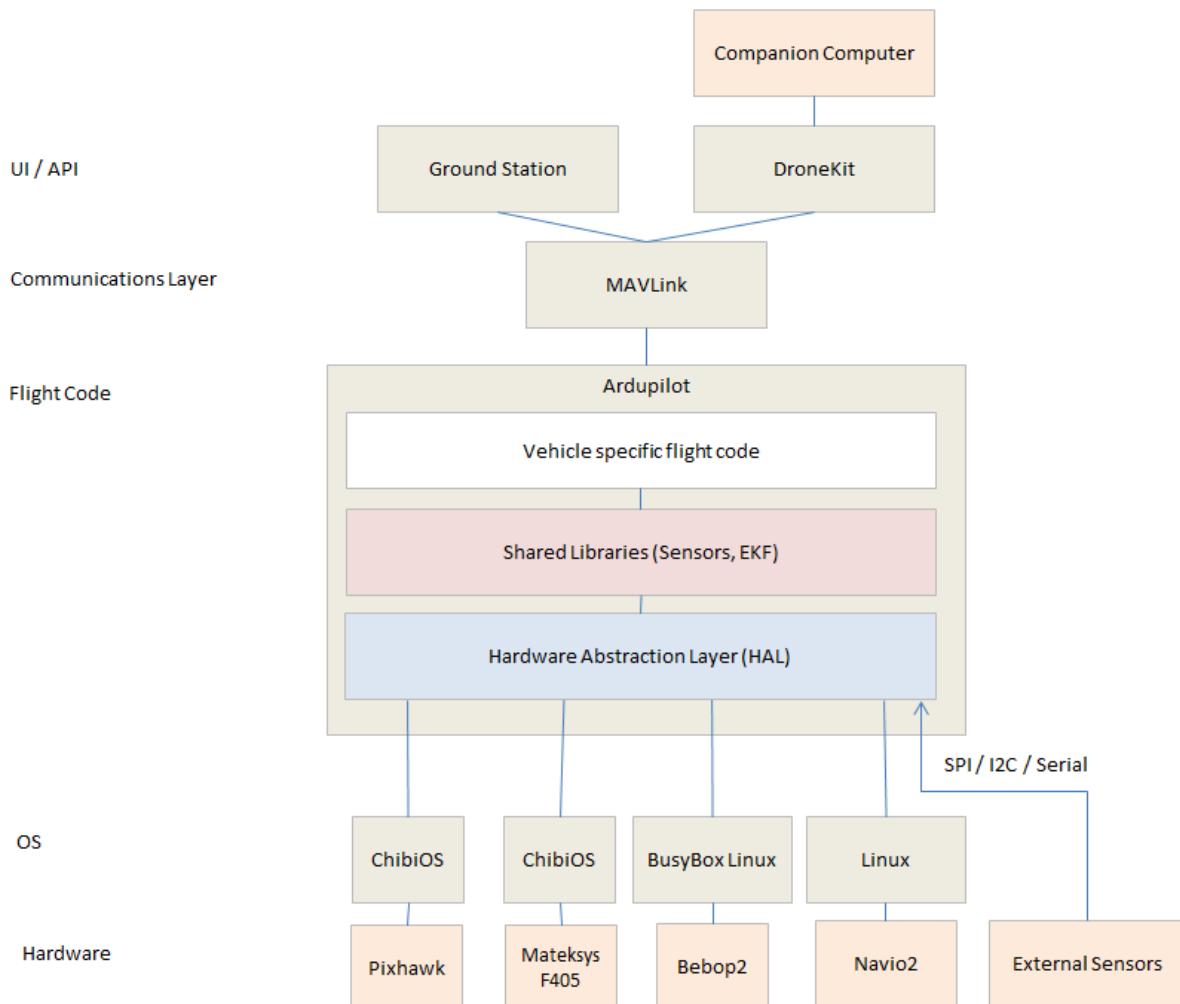


Figure 8.1: High Level Architecture of Ardupilot Software

8.2. 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 8.2. 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

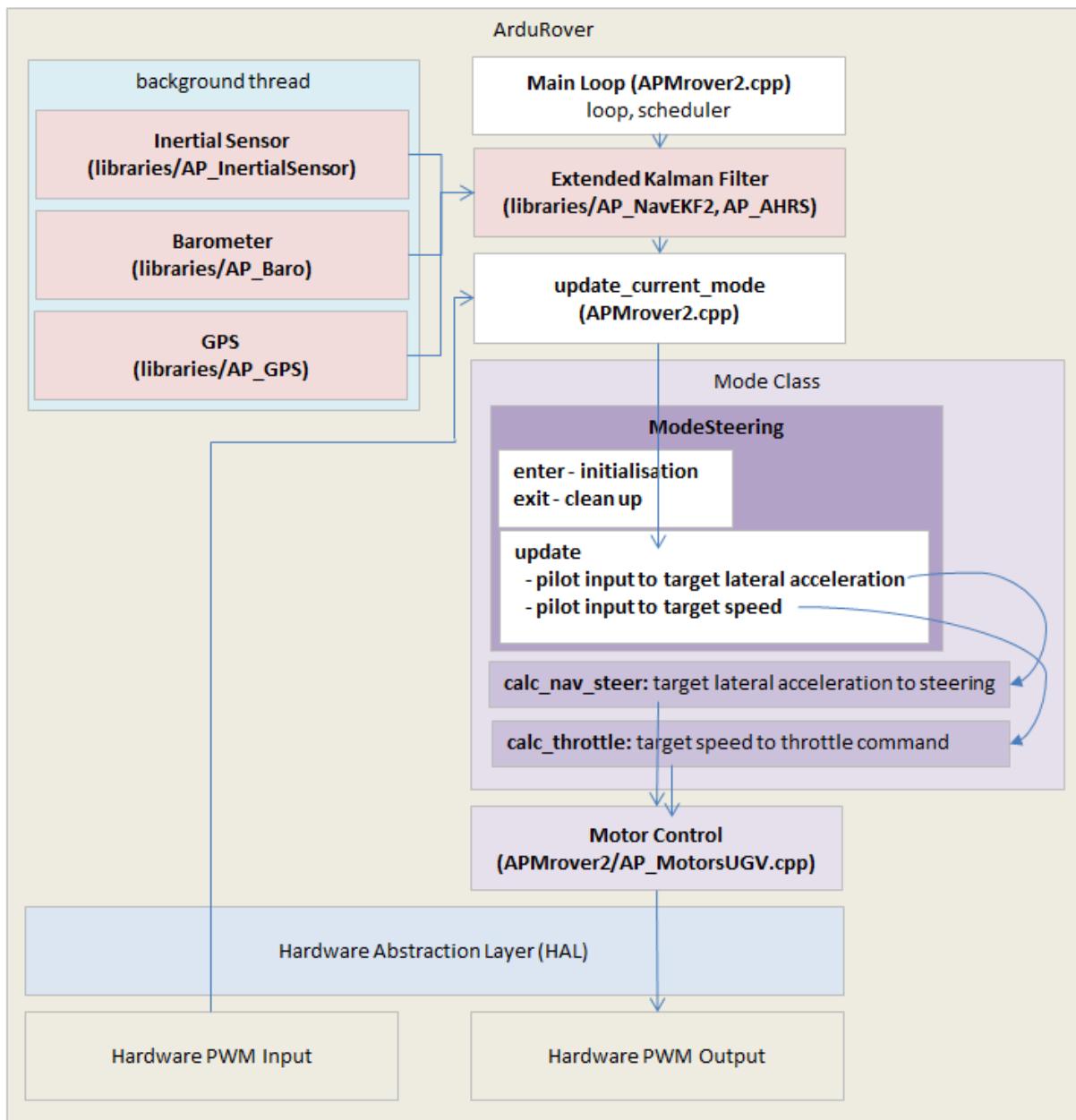
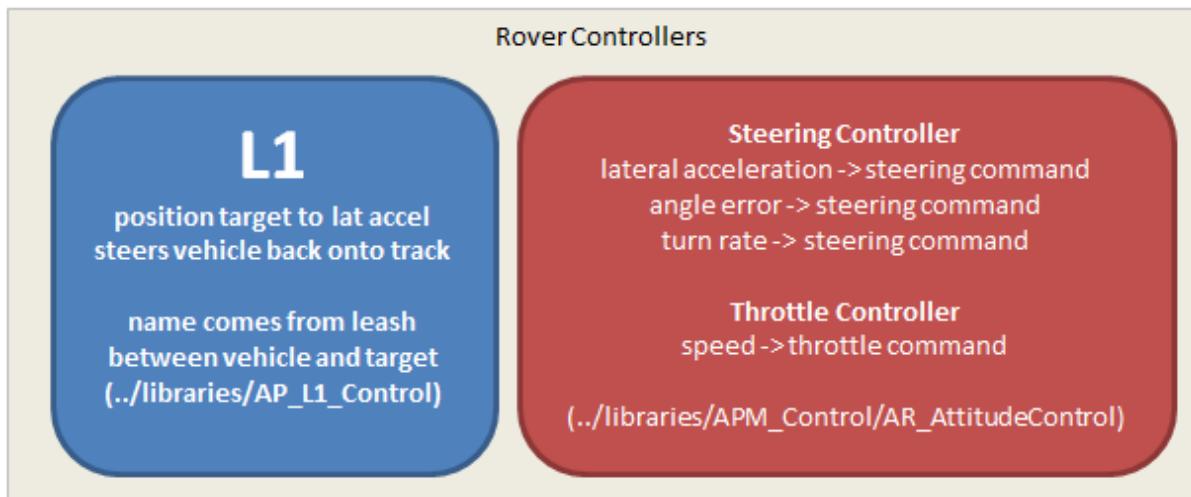


Figure 8.2: Rover architecture

The rover vehicle has three high level controllers, L1 controller, Steering Controller and Throttle controller. The three controllers are illustrated in Figure 8.3. 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 8.3:** 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.

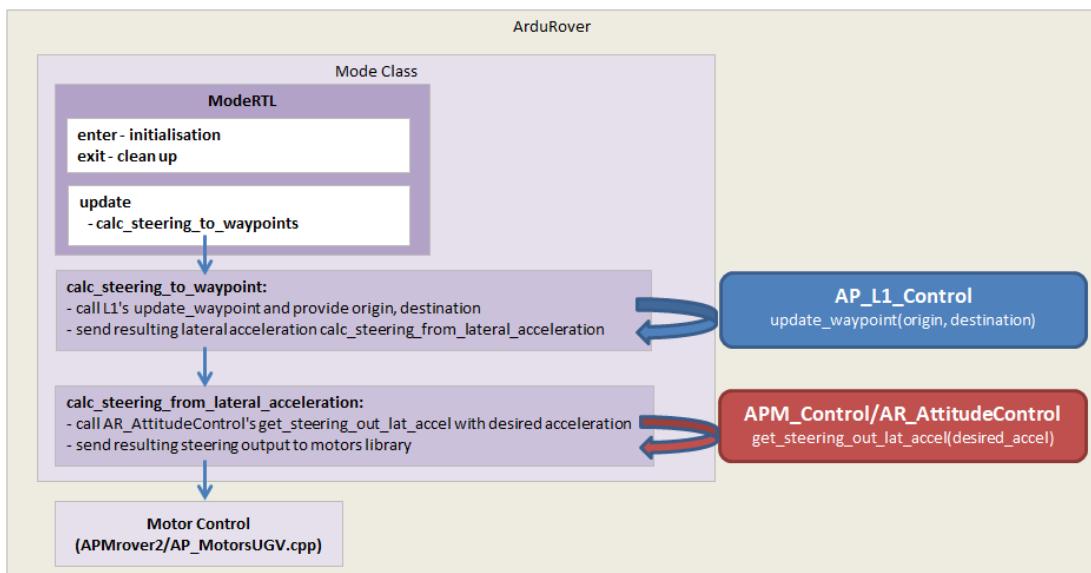
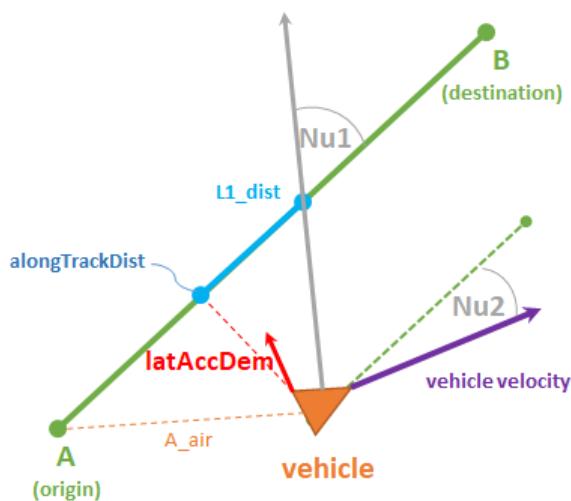
**Figure 8.4:** Rover navigation overview

Figure 8.4 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 8.5: Formulas for L1 controllers

8.3. 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 8.6. 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.

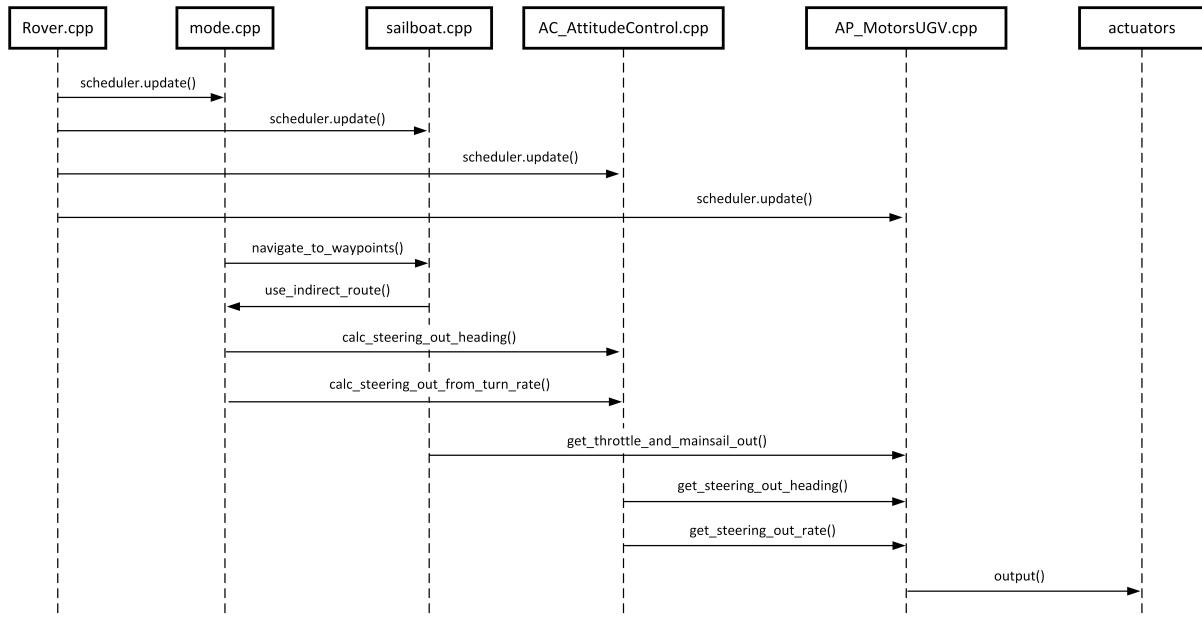


Figure 8.6: Code Execution Flow for Sailboat in auto mode

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.

8.3.1. The Shortcomings of Sailboat Framework

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

8.4. Ardupilots Control Strategies

8.4.1. 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 8.3: 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 [29], 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.

8.4.2. 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.

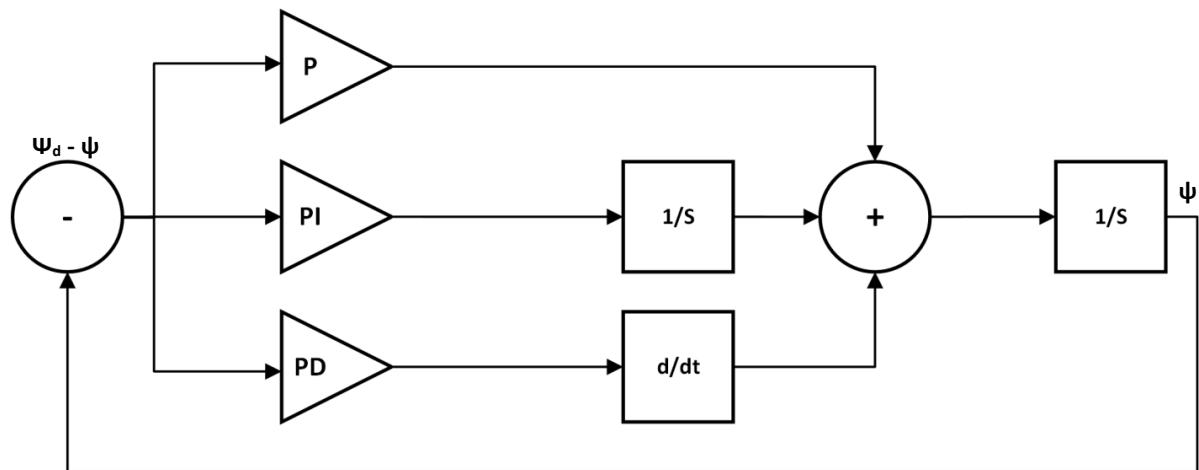


Figure 8.7: Ardupilots Rudder Controller

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.

8.4.3. 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.

8.4.4. 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.

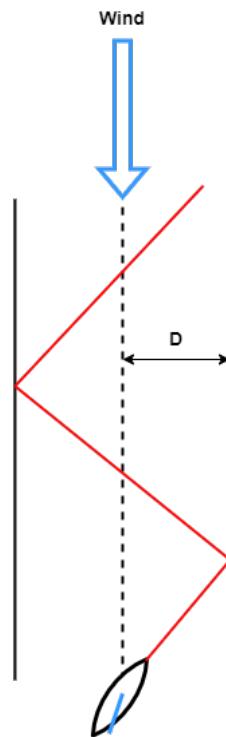


Figure 8.8: Tacking

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.

8.4.5. 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 8.9, 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.

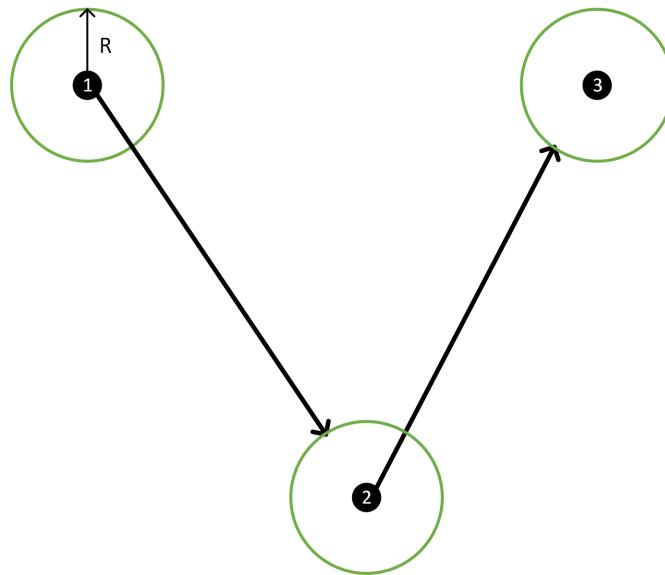


Figure 8.9: Ardupilots Waypoint Navigation

Chapter 9

Results

Chapter 10

Conclusion

Bibliography

- [1] R. van Tonder, “Github repository,” <https://github.com/22569596>, 2023.
- [2] M. Tranzatto, “Navigation and control for an autonomous sailing model boat,” Master’s thesis, University of Pisa, Lungarno Pacinotti 43, Italy, Sep. 2016.
- [3] Holybro, “Holybro pixhawk 6c,” http://docs.px4.io/main/en/flight_controller/pixhawk6c.html, 2023, pixhawk 6C.
- [4] L. Giger, S. Wismer, S. Boehl, G. A. Busser, H. Erckens, J. Weber, P. Moser, P. Schiwy, C. Pradalier, and R. Siegwart, “Design and construction of the autonomous sailing vessel avalon,” *Proceedings of The World Robotic Sailing Championship and International Robotic Sailing Conference*, July 2009.
- [5] 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.
- [6] K. L. Wille, “Autonomous sailboats,” Master’s thesis, Norwegian University of Science and Technology, Trondheim Ålesund Gjøvik, Norway, Jun. 2016.
- [7] “Df racing usa,” <https://www.dfracingusa.com/>, 2023, dF Racing.
- [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 m8n gps,” <https://holybro.com/collections/standard-gps-module>, 2023, m8N GPS.
- [11] Dronecode, “Mavlink protocol,” <https://mavlink.io/en/>, 2023, mAVLink.
- [12] 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.
- [13] T. I. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control*. John Wiley Sons Ltd, 2011.
- [14] K. D. Do and J. Pan, *Control of Ships and Underwater Vehicles*. Springer Science+Business Media, LLC, 2009.

- [15] 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.
- [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. L. Wille, V. Hassani, and F. Sprenger, “Modeling and course control of sailboats,” *International Federation of Automatic Control*, vol. 49, no. 23, pp. 532–539, 2016.
- [22] A. Gentry, “The aerodynamics of sail interaction,” *AIAA Symposium on the Aero/Hydronautics of Sailing*, vol. 1, no. 1, p. 8, 1971.
- [23] L. Larsson and R. E. Eliasson, *Principles of Yacht Design*. International Marine/McGraw-Hill, 2007.
- [24] K. Legursky, “System identification and the modeling of sailing yachts,” Ph.D. dissertation, University of Kansas, 2013.
- [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] L. Xianghui, B. U. Renxiang, and S. Wuchen, “Ship steering design based on nomoto equation,” *n Information Science and Control Engineering*, vol. 7, pp. 2175–2180, 2020.

- [28] W. da Silva Caetano and E. A. de Barros, “Identification of auv’s manoeuvrability using the nomoto’s first order equation,” *Symposium on Computing and Automation for Offshore Ship building*, vol. 2, no. 1, pp. 17–22, 2013.
- [29] D. H. dos Santos, “Framework for comparing autonomous sailboat control systems,” Ph.D. dissertation, Federak University of Rio Grando do Norte, 202.
- [30] J. A. A. Engelbrecht, “Advance automation 833: Introductory course to aircraft dynamics,” April 2019, aircrafts.
- [31] 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.
- [32] 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.
- [33] ArduPilot, “Ardupilot development,” <https://ardupilot.org/dev/index.html>, 2023, ardupilot.
- [34] D. T. Sen and T. C. Vinh, “Determination of added mass and inertia moment of marine ships moving in 6 degrees of freedom,” *International Journal of Transportation Engineering and Technology*, vol. 2, no. 1, pp. 8–14, April 2016.
- [35] J. M. J. Journée and L. J. M. Adegeest, “Theoretical manual of strip theory program “seaway for windows”,” <https://vdocument.in/seaway-theory-manual.html?page=12>, August 2003.
- [36] E. M. Lewandowski, *Modeling Software with finite State Machines*. World Scientific Publishing Co. Pte. Ltd., 2004.
- [37] D. Clarke, P. Gedling, and G. Hine, “The application of manoeuvring criteria in hull design using linear theory,” *Transactions of the Royal Institution of Naval Architects, RINA*., 1982.
- [38] G. F. Franklin, J. D. Powell, and A. Emami-Naeini, *Feedback Control of Dynamic Systems*. Pearson, 2020.

Appendix A

Sailboat Specifications

This appendix provides the methods/equations used to gain model specifications on the sailboat. These specification are used for the model of the sailboat and in the design of the control systems.

A.1. Sailboat Specifications

The sailboat specifications are shown in Table A.1,

Table A.1: Sailboat Geometry

Meaning	Symbol	Value
Length	L	0.95
Breadth	B	0.125
Draft	T	0.12
Rig Height	H_r	0.105
Overall Height	H_o	1.470
Mass	m	2
Hull Speed	h_s	1.3
Overall Sail Area	A_s	0.3736
Water friction coefficient	p_1	0.3736
Water friction coefficient	p_2	0.3736
Water angular friction coefficient	p_3	0.3736
Lift coefficient of the sail	p_4	0.3736
Lift coefficient of the rudder	p_5	0.3736
Distance between the mast and the CoE of the sail	p_6	0.3736
Distance between the boat's centre of gravity and the mast	p_7	0.3736
Distance between G and the rudder	p_8	0.3736
Roll friction coefficient	p_9	0.3736
Length of the equivalent pendulum in roll motion	p_{10}	0.3736

A.2. Determination of Added Mass and Inertia Moment of the sailboat

The approximate added mass and moment of inertia values for the sailboat is calculated using the methods formulated and reviewed in [34]. The method used is known as the method of equivalent ellipsoid. For a ocean vessel, the most representative shape of the hull is a elongated ellipsoid, as illustrated in Figure A.1, with $c/b = 1$ and $r = a/b$. Where a, b are semi axis of the ellipsoid

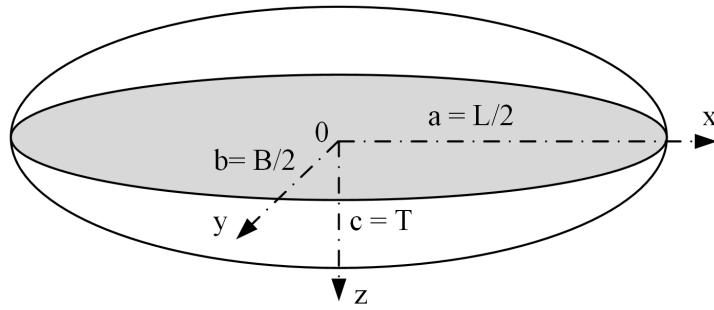


Figure A.1: Vessel assumed as an ellipsoid

Based on this method the following components of added mas can be calculated, m_{11} , m_{22} , m_{33} , m_{44} , m_{55} and m_{66} . The components are described by the following equations,

$$m_{11} = mk_{11} \quad (\text{A.1})$$

$$m_{22} = mk_{22} \quad (\text{A.2})$$

$$m_{33} = mk_{33} \quad (\text{A.3})$$

$$m_{44} = k_{44}I_{xx} \quad (\text{A.4})$$

$$m_{55} = k_{55}I_{yy} \quad (\text{A.5})$$

$$m_{66} = k_{66}I_{zz} \quad (\text{A.6})$$

Each calculation make use of a term k_{ij} , called hydrodynamic coefficient, these coefficient is calculated as,

$$k_{11} = \frac{A_0}{2 - A_0} \quad (\text{A.7})$$

$$k_{22} = \frac{B_0}{2 - B_0} \quad (\text{A.8})$$

$$k_{33} = \frac{C_0}{2 - C_0} \quad (\text{A.9})$$

$$k_{44} = 0 \quad (\text{A.10})$$

$$k_{55} = \frac{(L^2 - 4T^2)^2(A_0 - C_0)}{2(c^4 - a^4) + (C_0 - A_0)(4T^2 + L^2)^2} \quad (\text{A.11})$$

$$k_{66} = \frac{(L^2 - B^2)^2(B_0 - A_0)}{2(L^4 - B^4) + (A_0 - B_0)(L^2 + B^2)^2} \quad (\text{A.12})$$

where A_0 and B_0 is calculated by,

$$A_0 = \frac{2(1 - e^2)}{e^3} \left[\frac{1}{2} \ln \left(\frac{1 + e}{1 - e} \right) - e \right] \quad (\text{A.13})$$

$$B_0 = C_0 = \frac{1}{e^2} - \frac{1-e^2}{2e^3} \ln \left(\frac{1+e}{1-e} \right) \quad (\text{A.14})$$

with,

$$e = \sqrt{1 - \frac{b^2}{a^2}} = \sqrt{1 - \frac{d^2}{L^2}} \quad (\text{A.15})$$

d and L are the diameter and length of the vessel respectively. The moment of inertia of the displaced water is approximately the moment of inertia of the ellipsoid,

$$I_{xx} = \frac{1}{120} \pi \rho L B T (4T^2 + B^2) \quad (\text{A.16})$$

$$I_{yy} = \frac{1}{120} \pi \rho L B T (4T^2 + L^2) \quad (\text{A.17})$$

$$I_{zz} = \frac{1}{120} \pi \rho L B T (B^2 + L^2) \quad (\text{A.18})$$

The missing added mass components, m_{42} , m_{24} , m_{26} and m_{62} , are calculated in [35].

In [36] it is shown that the missing added mass coefficients can be approximated as zero-frequency roll motions which is generally acceptable in maneuvering motions, and is equal to,

$$m_{26} = m_{62} = \frac{A_{44}(x)}{\rho \pi T(x)^4 / 8} = H^4 \left(\frac{128}{\pi^2} \frac{a^2(1+b)^2 + \frac{8}{9}ab(1+b) + \frac{16}{9}b^2}{(1+a+b)^4} \right) \quad (\text{A.19})$$

$$m_{24} = m_{42} = \frac{A_{24}(x)}{\rho \pi T(x)^3 / 2} = -\frac{16}{3\pi} \left(\frac{a(1-a+\frac{4}{5}b-ab+\frac{3}{5}b^2) + \frac{4}{5}b - \frac{12}{7}b^2}{((1-a)^2+3b^2)(1-a+b)} \right) \quad (\text{A.20})$$

where a and b is calculated as,

$$a = (b+1)q \quad (\text{A.21})$$

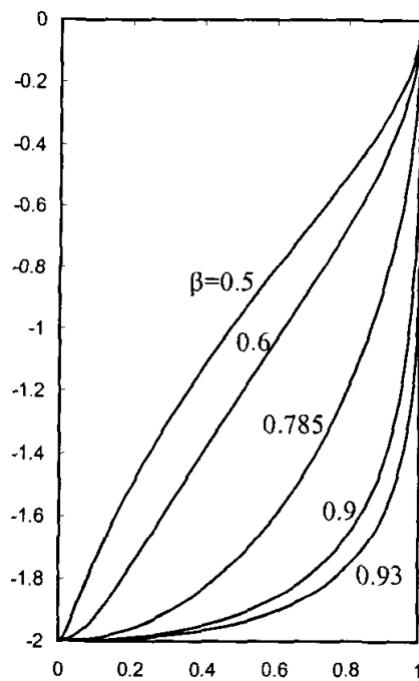
$$b = \frac{\frac{3}{4}\pi + \sqrt{(\frac{\pi}{4})^2 - \frac{\pi}{2}p(1-q^2)}}{\pi + p(1-q^2)} - 1 \quad (\text{A.22})$$

with p and q is,

$$q = \frac{H-1}{H+1} \quad (\text{A.23})$$

$$p = \beta - \frac{\pi}{4} \quad (\text{A.24})$$

To determine H and β the hull shape of the Dragonflight 95 was inspected. The ratio of H and β represents the shape of the hull. An example of a couple of hull shapes is illustrated in Figure A.2.

**Figure A.2:** Hull form with $H = 0.5$

The sailboat's added mass and moment of inertia values are shown in Table A.2.

Table A.2: Sailboat Added Mass and Moment of Inertia

Symbol	Value
m_{11}	0.06328
m_{22}	1.79327
m_{33}	
m_{44}	0
m_{55}	545
m_{66}	1.01366
m_{24}	0.628
m_{42}	0
m_{26}	0
m_{62}	0
I_{xx}	0.81153
I_{yy}	3.58179
I_{zz}	1.14753

A.3. Determination of Linear Damping Forces and Moments

The linear damping coefficients, expressed in [36] are also severed to as the stability derivatives since they dictate the course-keeping stability of the vessel. The equations presented below only gives a crude approximation of the forces and should be used with care. In this project the values were only used for simulation purposes and not for designing the steering controller. The linear damping are modelled as,

$$Y_v = \frac{dY}{dx} = -\frac{1}{2}\rho U(LT) \frac{\pi T}{L} \quad (\text{A.25})$$

$$N_v = \frac{dN}{dv} = \frac{L}{2} Y_v = -\frac{1}{2}\rho U(LT) \frac{\pi T}{L} \frac{L}{2} \quad (\text{A.26})$$

$$Y_r = \frac{dY}{dr} = \frac{1}{4}\pi\rho U T^2 \quad (\text{A.27})$$

$$N_r = \frac{dN}{dr} = -\frac{1}{8}\pi\rho U T^2 L \quad (\text{A.28})$$

The linear damping forces was later revised in [37], resulting in the following forces normalized around

$$\frac{1}{2}\rho U^2 L^2 \quad (\text{A.29})$$

The normalized linear coefficients are,

$$Y'_v = -\pi \frac{T^2}{L^2} \quad (\text{A.30})$$

$$N'_v = -\frac{1}{2}\pi \frac{T^2}{L^2} \quad (\text{A.31})$$

$$Y'_r = \frac{1}{2}\pi \frac{T^2}{L^2} \quad (\text{A.32})$$

$$N'_r = -\frac{1}{4}\pi \frac{T^2}{L^2} \quad (\text{A.33})$$

The revised normalized coefficients are,

$$Y'_{vH} = -\pi \frac{T^2}{L^2} \left(1 + 0.4C_B \frac{B}{T} \right) \quad (\text{A.34})$$

$$N'_{vH} = -\pi \frac{T^2}{L^2} \left(0.5 + 2.4 \frac{T}{L} \right) \quad (\text{A.35})$$

$$Y'_{rH} = \pi \frac{T^2}{L^2} \left(0.5 - 2.2 \frac{B}{L} + 0.08 \frac{B}{T} \right) \quad (\text{A.36})$$

$$N'_{rH} = -\pi \frac{T^2}{L^2} \left(0.25 + 0.039 \frac{B}{T} - 0.56 \frac{B}{L} \right) \quad (\text{A.37})$$

The sailboat's linear damping coefficient values are shown in Table A.3.

Table A.3: Sailboat Linear Damping Coefficients

Symbol	Value
Y_v	6
N_v	6
Y_r	6
N_r	6

A.4. Identification of Nomoto's First Order System

The first order system of Nomoto is identified using Kempf's Zig-Zag Maneuver [28]. The Zig-Zag maneuver with notations is illustrated in Figure A.3. The procedure is as follows:

1. Adjust the rudder to 20 degrees
2. Wait for the yaw angle to reach 20 degrees
3. Adjust the rudder to -20 degrees
4. Wait for the yaw angle to reach -20 degrees
5. Repeat from 1.

It is required that the initial conditions are as follows,

$$\Psi = 0 \quad (\text{A.38})$$

$$\dot{\Psi} = 0 \quad (\text{A.39})$$

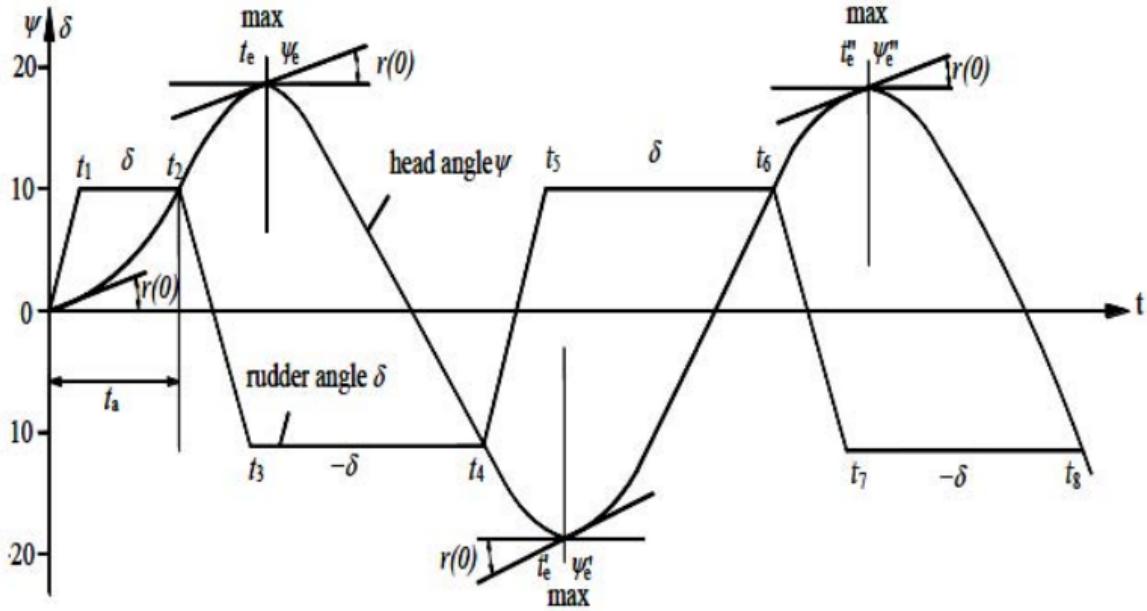


Figure A.3: Notations used for the analysis of zig-zag maneuver

The first order system Nomoto system is therefore equal to,

$$T\Psi + \Psi = K\delta_r t + K \int_0^t \delta_m(t) dt \quad (\text{A.40})$$

The values for K can thus be calculated as,

$$K = \frac{\Psi_e}{\delta_r t_e + \int_0^{t_e} \delta_m(t) dt} \quad (\text{A.41})$$

Once K is calculated the value for t_e , t'_e and t''_e . The appropriate value for T for each k can be calculated by apply $t = t_0$, $t = t'_0$ and $t = t''_0$ to Equation A.41 to obtain:

$$T = \frac{K}{\dot{\Psi}(t_0)} \left(\int_0^{t_0} \delta_m(t) dt + \delta_r t_0 \right) \quad (\text{A.42})$$

$$T = \frac{K}{\dot{\Psi}(t'_0)} \left(\int_0^{t'_0} \delta_m(t) dt + \delta_r t'_0 \right) \quad (\text{A.43})$$

$$T = \frac{K}{\dot{\Psi}(t''_0)} \left(\int_0^{t''_0} \delta_m(t) dt + \delta_r t''_0 \right) \quad (\text{A.44})$$

Due to $\dot{\Psi} = 0$ the integral term can be calculated as,

$$(4) = \int_0^t \delta_m(t) dt = \delta_1 \left(-\frac{t_1}{2} + \frac{1}{2}(t_2 + t_3) \right) + \delta_2 \left(-\frac{1}{2}(t_2 + t_3 + t) \right) \quad (\text{A.45})$$

$$(6) = \int_0^t \delta_m(t)dt = \delta_1 \left(-\frac{t_1}{2} + \frac{1}{2}(t_2 + t_3) \right) + \delta_2 \left(-\frac{1}{2}(t_2 + t_3 + t) \right) + \delta_3 \left(-\frac{1}{2}(t_4 + t_5) + t \right) \quad (\text{A.46})$$

$$(8) = \int_0^t \delta_m(t)dt = \delta_1 \left(-\frac{t_1}{2} + \frac{1}{2}(t_2 + t_3) \right) + \delta_2 \left(-\frac{1}{2}(t_2 + t_3 + t) \right) \\ + \delta_3 \left(-\frac{1}{2}(t_4 + t_5) + t \right) + \delta_4 \left(-\frac{1}{2}(t_6 + t_7) + t \right) \quad (\text{A.47})$$

Where 4, 6 and 8 are illustrated in Figure A.3 as the different Zig-Zag maneuvers. The values for k_6 and k_8 is also calculated with the same equation just by swapping the integral term. The calculated values for the gain and time constant is shown in Table A.4.

Table A.4: Sailboat K and T Values

Symbol	Value
K	-
T	-

Appendix B

Additional Modelling Information

B.1. Modeling

B.1.1. Rigid-Body System Inertia Matrix \mathbf{M}_{RB}

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

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

where

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

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.

B.1.2. 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,

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

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,

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

Appendix C

Classical Control Systems Theory

The classical control systems theory discussed in this Chapter is proved and formulated in [38]. The two parts of theory discussed in this Chapter are first the time-domain specifications and secondly the root locus design method.

C.1. Time-Domain Specifications

These specifications are used to full fill the design specifications and calculated the required closed-poles in the s-domain. The requirements of a step response are expressed in terms of standard quantities in Figure C.1.

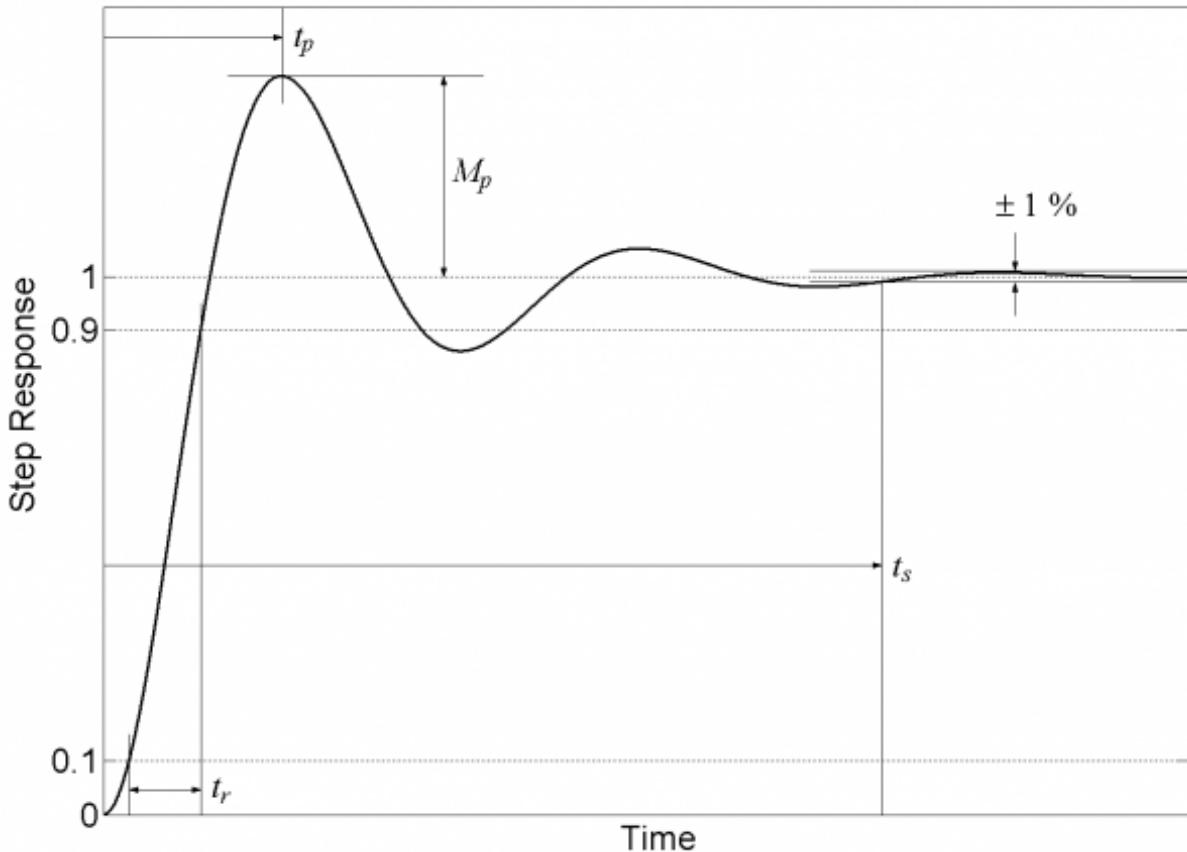


Figure C.1: Definition of time-domain specifications

The rise time t_r , defined as the time a system takes to go from 0.1 to 0.9 of the finale

value, can be calculated by,

$$t_r = \frac{1.8}{\omega_n} \quad (\text{C.1})$$

It is important to note that this is only an approximation and becomes less accurate once zeros are introduced to a system. The peak time t_p , the time the system takes to reach its peak value, is calculated by,

$$t_p = \frac{\pi}{\omega_d} \quad (\text{C.2})$$

The overshoot M_p , defined as the percentage of the step response value that the system overshoots, is calculated by,

$$M_p = e^{-\pi\zeta/\sqrt{1-\zeta^2}} \quad (\text{C.3})$$

When the overshoot is part of the design requirements and damping coefficient, ζ , needs to be calculated the following equation is used,

$$\zeta = \sqrt{\frac{\log(M_p)^2}{pi^2 + \log(M_p)^2}} \quad (\text{C.4})$$

The settling time t_s , defined as the time it takes the system to be within 5 percent of the final value, can be calculated by,

$$t_s = \frac{4}{\omega_n} \quad (\text{C.5})$$

Generally in designing a step response the values for t_r , t_s and M_p is specified and the desired pole pair is calculated from the equations above.

C.2. Root Locus Design

The root locus can roughly be described as a function that presents the possible closed-poles as a function of gain of the system. The root locus is drawn from the characteristic equation. Figure C.2 illustrates a basic closed-loop system's block diagram. The transfer function of the closed-loop system is defined as,

$$\frac{Y(s)}{R(s)} = \frac{D(s)G(s)}{1 + D(s)G(s)H(s)} \quad (\text{C.6})$$

The characteristic equation, which is the poles of the transfer function is,

$$1 + D(s)G(s)H(s) = 0 \quad (\text{C.7})$$

The characteristic equation is changed to the following form to be more suitable for the root locus design.

$$1 + KL(s) = 0 \quad (\text{C.8})$$

where K is the varying gain.

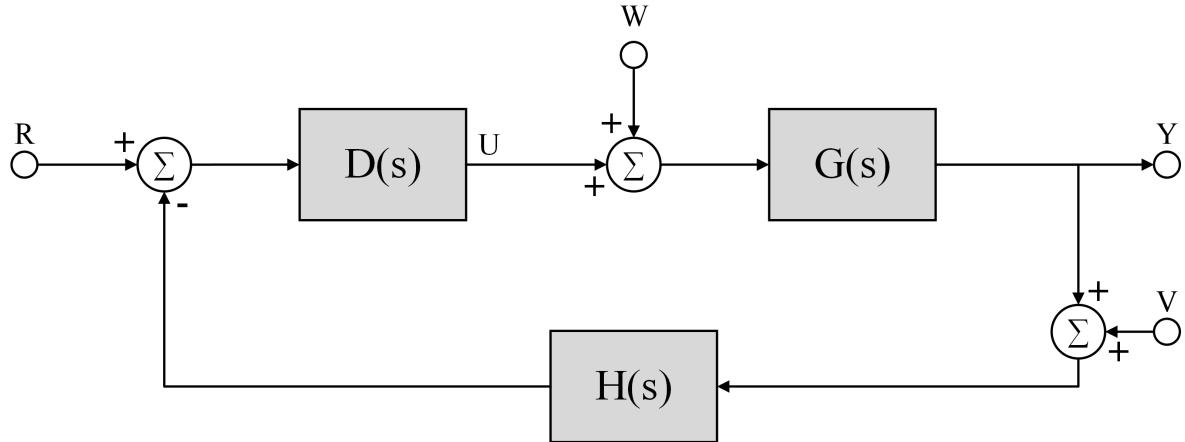


Figure C.2: Basic closed-loop block diagram

The following definitions are used to calculate the and graph a root locus, these definitions are used in designing controllers to force the closed-loop poles to their desired locations.

Definition 1

The root locus is the set of values of s for which $1 + KL(s) = 0$ is satisfied as the real parameter K varies from 0 to $+\infty$. The roots on the locus are the closed-loop poles of the system.

Definition 2

The root locus of $L(s)$ is the set of points in the s-plane where the phase of $L(s)$ is 180° . The angle from a zero to the desired pole is defined as ψ_i and the angle from a pole to the desired pole is defined as ϕ_i . The definition 2 is expressed as,

$$\sum \psi_i - \sum \phi_i = 180^\circ + 360^\circ(l - 1) \quad (\text{C.9})$$

Definition 3

To determine the value for K such that the desired closed-loop poles are achieved, the magnitude definition is used.

$$K = -\frac{1}{L(s)} = \frac{1}{|L|} \quad (\text{C.10})$$

where $|L|$ is the distance between the desired poles and the zeros and poles of the system.

Appendix D

Additional Platform Development

D.1. 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

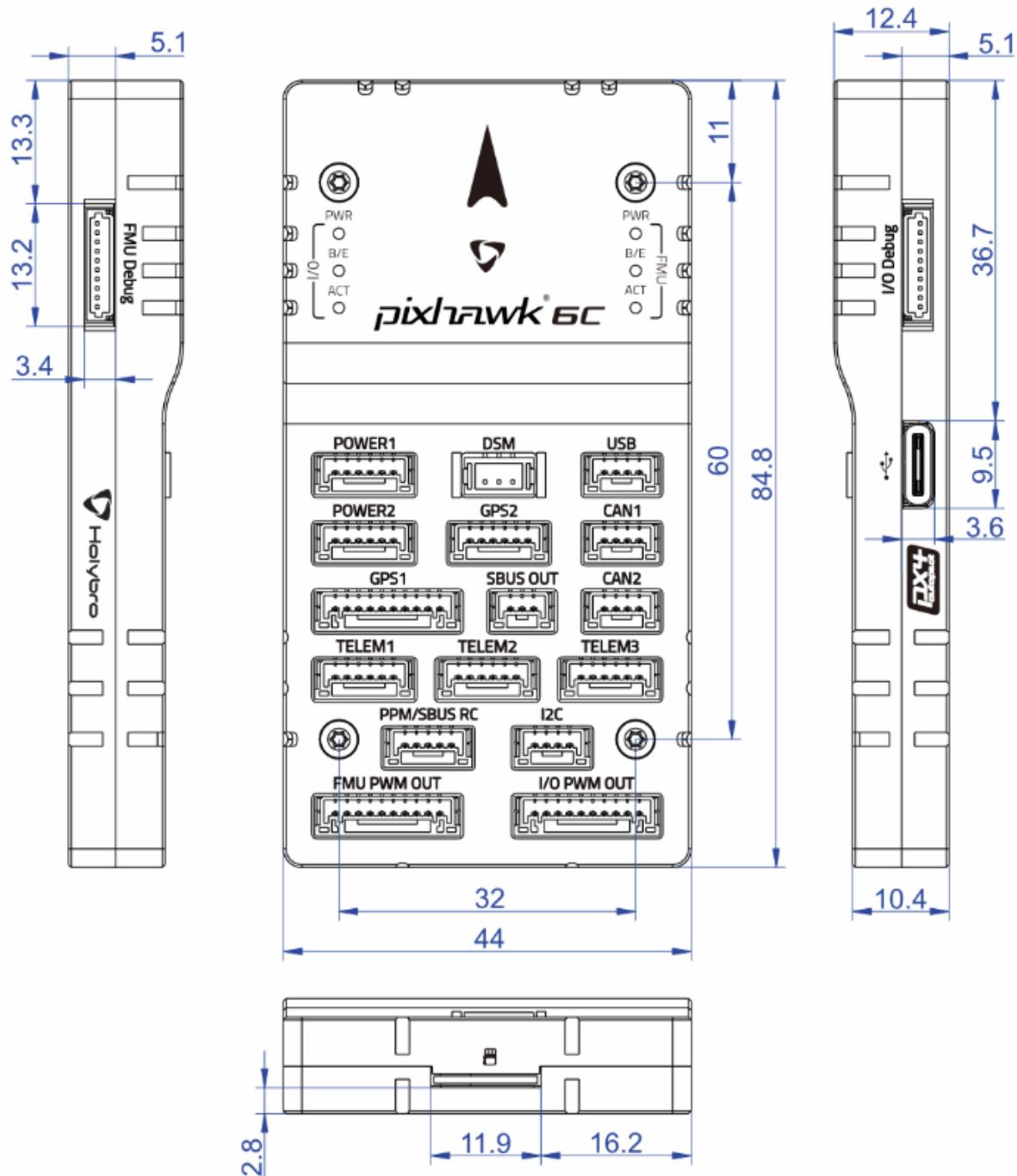


Figure D.1: Pixhawk 6C Dimensions

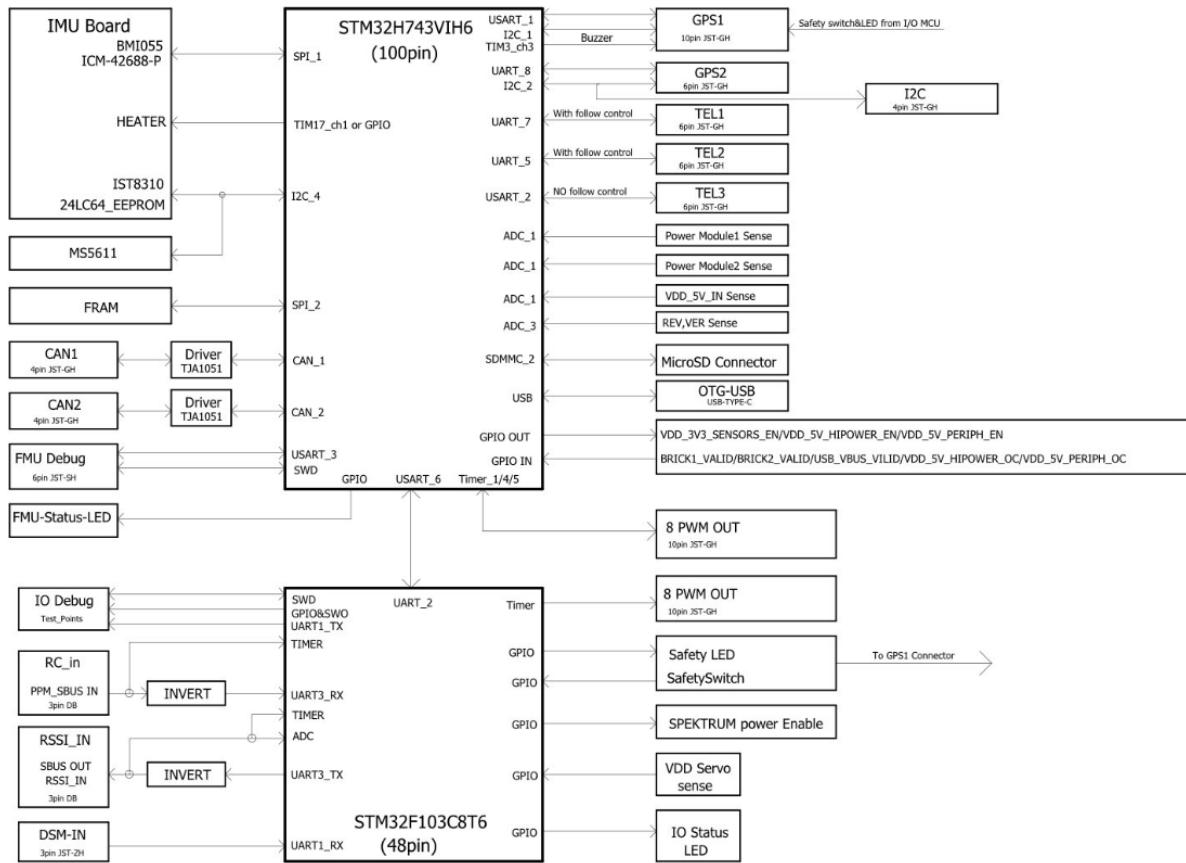


Figure D.2: Pixhawk 6C Pinout Diagram

D.2. 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

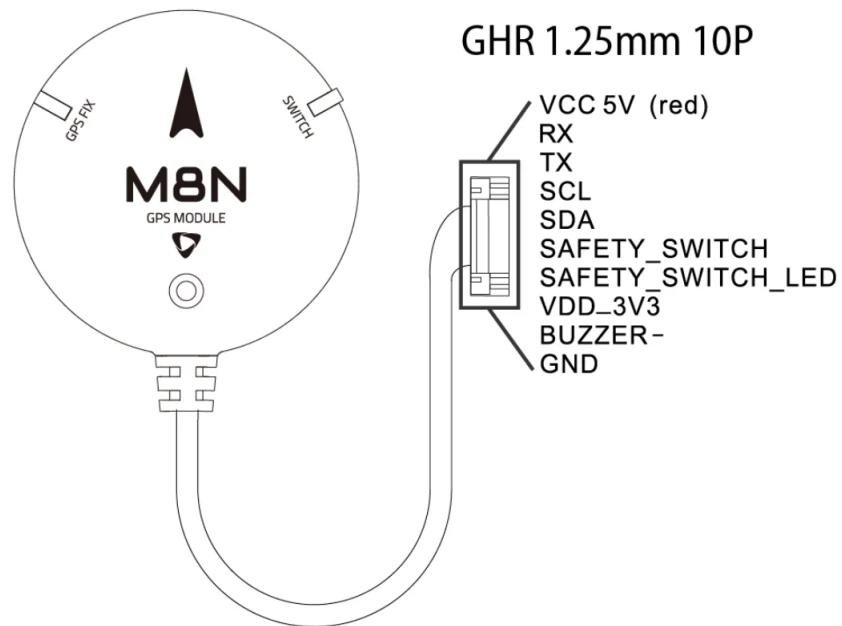


Figure D.3: M8N GPS pinout diagram

D.3. Ardupilots Sailboat Configuration Parameters

Table D.1: 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 D.1: 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

D.4. Copter Modes Architecture

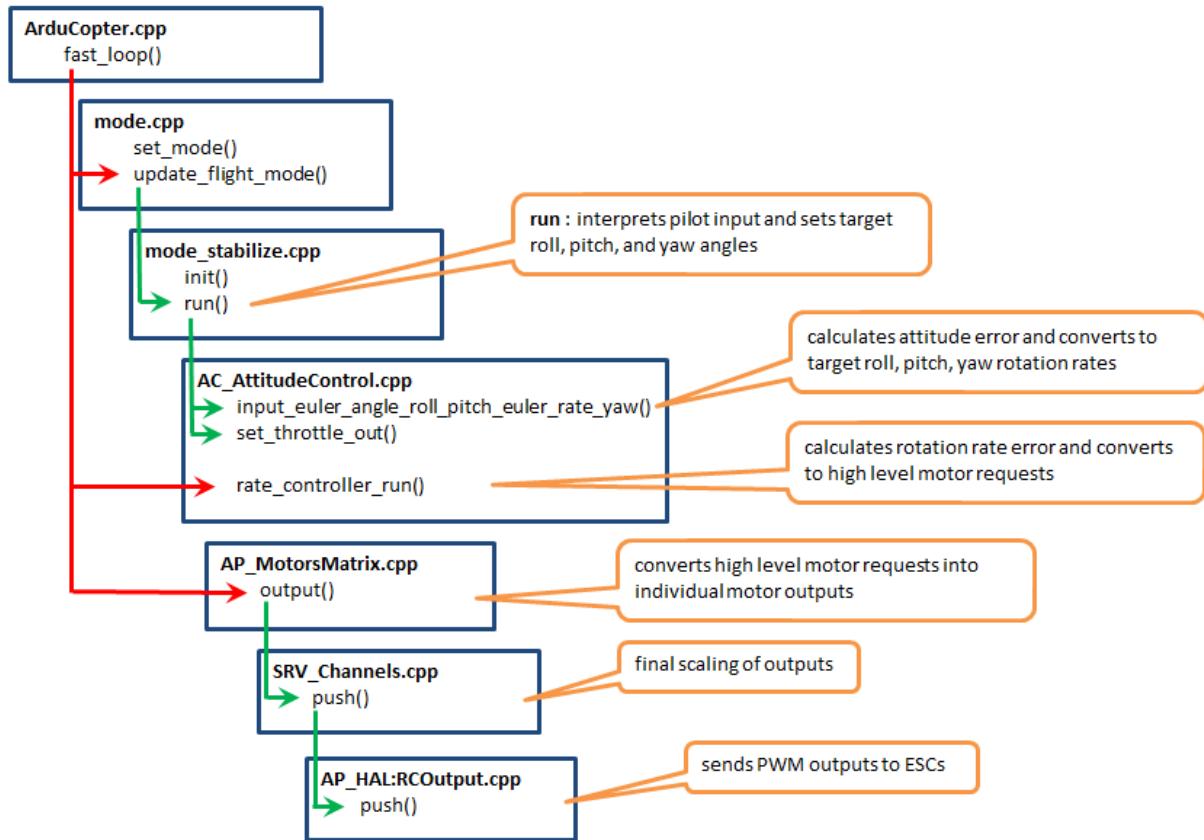


Figure D.4: Manual Flight Mode

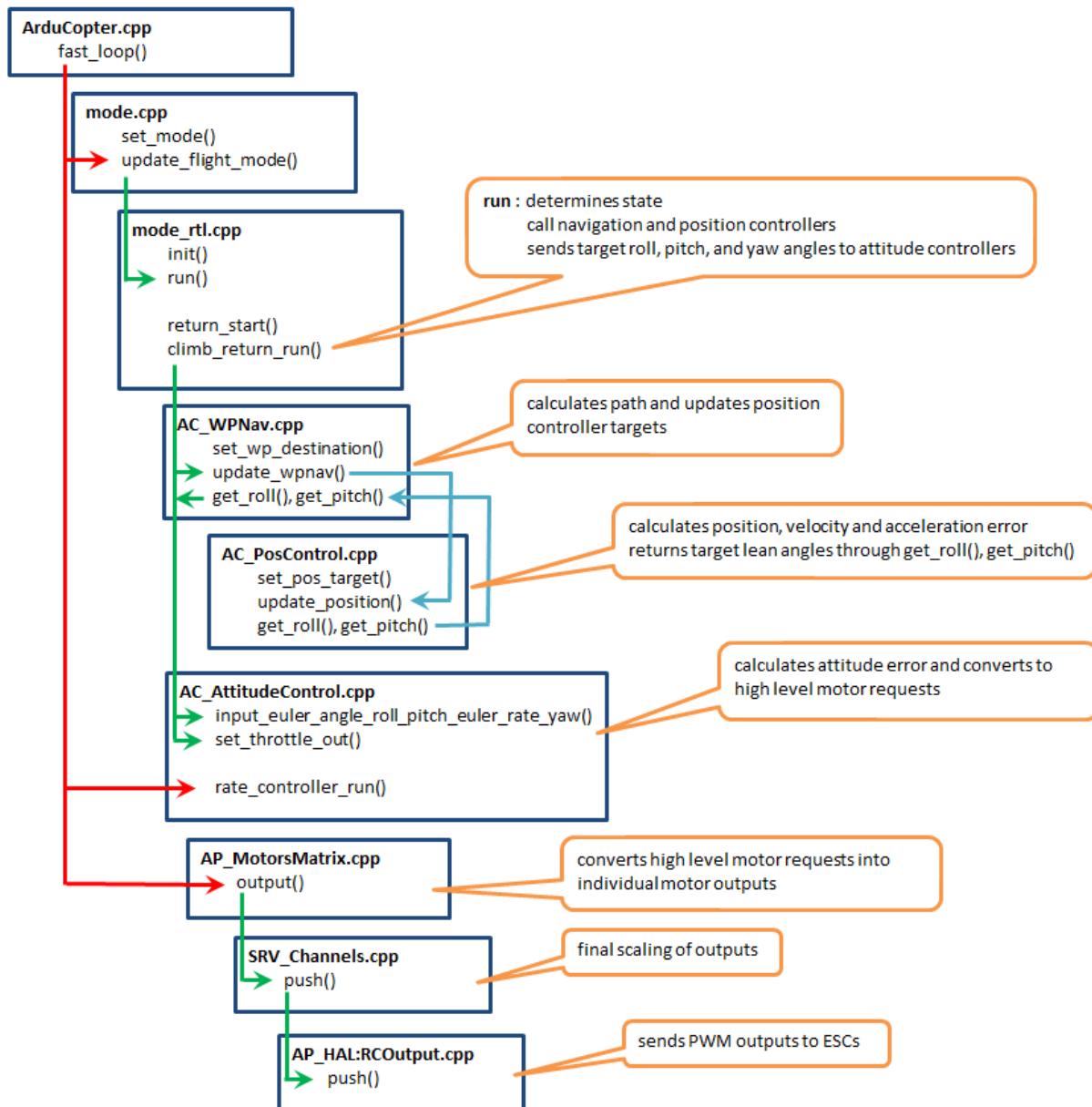


Figure D.5: Autonomous Flight Mode

D.5. 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 D.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 D.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 D.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 D.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 }
```