

Master's thesis

NTNU
Norwegian University of Science and Technology
Department of Engineering Cybernetics

Jørgen Mikal Benum

Autonomous saildrone simulation and motion control

Master's thesis in Cybernetics and Robotics
Supervisor: Morten Dinhoff Pedersen
June 2022



Norwegian University of
Science and Technology

Jørgen Mikal Benum

Autonomous saildrone simulation and motion control

Master's thesis in Cybernetics and Robotics

Supervisor: Morten Dinhoff Pedersen

June 2022

Norwegian University of Science and Technology
Department of Engineering Cybernetics



Kunnskap for en bedre verden

Preface

First, I would like to thank my supervisor for all the help; the provided note on convexification by tacking and engaging discussions on various topics within this thesis.

I want to give a special thanks to my family and my girlfriend for helping me stay motivated through all the hard work. Finally, I want to give a special thanks to my grandfather, who passed away lately. He has always been an instrumental and inspiring person in my life, and I would like to thank him for all the good memories.

Abstract

In this thesis, a motion control system for a sailboat drone is developed and tested for the purpose of collision avoidance and autonomous path following under various environmental conditions. In addition, a simulation framework is developed, consisting of a comprehensive model of the system dynamics and a simplified perception model.

Sail propulsion enables long mission duration for autonomous crafts, but also represents system restrictions as the propulsion is dependent on the wind. The ocean current, wind and sail combined impose constraints on the obtainable velocities along different course directions, as well as which maneuvers are feasible. These velocities and courses are often summarized in a feasible velocity diagram used for course selection. Where an effective velocity in a non-feasible direction is obtained by periodic switching between two feasible directions.

The constraint results in a non-trivial motion control problem which becomes even more complex when incorporating collision avoidance. Multiple approaches to solve the motion control problem exists, often exploiting a feasible velocity diagram together with an automatic control hierarchy. However, the approaches found neglect the effect of ocean current, and only a few studies are concerned with collision avoidance and feasibility of maneuvers.

To address this, a motion control system has been developed. The system exploits a proposed path following algorithm, which switches between line-of-sight guidance and course control. The algorithm builds on a derived method to incorporate ocean current into the feasible velocity diagram, enabling the system to provide a course reference that is optimal with respect to both current and wind. Further, the system contains a proposed collision avoidance algorithm, where obstacles are introduced as velocity vector constraints. In addition, the forward propulsion induced by the sail is optimized offline by exploiting the lift and drag coefficient models, forming a lookup-table for the optimal angle of attack for the sail.

The motion control system was implemented and tested in the simulation framework to evaluate the system's performance as a whole under various environmental conditions. The simulation result shows that the sailboat can follow the desired path even under harsh environmental conditions. The results also reveal that incorporating ocean current into course selection can yield significantly higher effective velocity and increase robustness. In addition, static obstacles are successfully avoided, even when the direction of travel requires the composition of two courses.

Sammendrag

I denne oppgaven er et automatisk kontrollsysteem for en seilbåt drone utviklet og testet med det formål å unngå kollisjon, og å oppnå autonom banefølging under ulike miljøforhold. I tillegg er det utviklet et simuleringsrammeverk bestående av en modell av seilbåtdronens systemdynamikk og en forenklet oppfatningsmodell av omgivelsene.

Seilfremdrift muliggjør lang oppdragsvarighet for autonome fartøyer, men representerer også en systembegrensning da fremdriften er avhengig av vinden. Havstrømmen, vinden og seilet kombinert setter begrensninger på de oppnåelige hastighetene langs de ulike kursretningene, samt hvilke manøvrer som er mulige. Hastighetene og kursretningene er ofte oppsummert i et hastighetsdiagram som brukes for kursvalg. I enkelte kursretninger vil det ikke være mulig å seile direkte, slik at en sammensetning av to alternative kursretninger må benyttes for å oppnå en effektiv hastighet i den ønskede retningen.

Begrensningen resulterer i et ikke-triviert styringsproblem som blir enda mer komplekst når man inkluderer kollisjonsunngåelse. Det er benyttet flere ulike metoder for å løse styringsproblemet. Ofte benyttes et hastighetsdiagram sammen med et automatisk kontrollsysteem. I tidligere løsninger neglisjeers ofte effekten av havstrømmen. I tillegg har lite arbeid blitt rettet mot kollisjonsunngåelse og manøver valg ved lave hastigheter.

For å håndtere styringsproblemet er det i denne oppgaven utviklet et automatisk kontrollsysteem. Systemet benytter en algoritme for banefølging som velger mellom siktlinjeveiledning og kurskontroll. Algoritmen bygger på en utledet metode for å inkludere havstrøm i hastighetsdiagrammet, slik at systemet kan gi en kurssreferanse som er optimal med hensyn på både strøm og vind. Videre inneholder systemet en kollisjonsunngåelsesalgoritme, der hindringer introduseres som hastighetsvektorbegrensninger. I tillegg er fremdriften gitt av seilet optimalisert ved å benytte seilets løft og drag modeller. Dette resulterer i en oppslagstabell over optimale angrepsvinkler for seilet.

Det automatiske kontrollsystemet er implementert og testet i simuleringsrammeverket for å evaluere systemets ytelse som helhet under ulike miljøforhold. Simuleringsresultatet viser at seilbåten kan følge den ønskede banen selv under tøffe miljøforhold. Resultatene viser også at å hensynta havstrøm i kursvalg kan gi betydelig høyere effektiv hastighet og øke robustheten. I tillegg unngår statiske hindringer, selv når den effektive kursretningen krever sammensetning av to kurser.

Contents

1	Introduction	1
1.1	Background and motivation	1
1.2	Previous and related work	2
1.3	Problem description	4
1.4	Approach	4
1.5	Contributions	5
1.6	Limitations	5
1.7	Assumptions	6
1.8	Outline	6
2	Background and essentials	7
2.1	System description and definitions	7
2.2	Terminology and sailing maneuvers	8
2.3	Modeling preliminaries and definitions	10
2.3.1	Reference frames and definitions	10
2.3.2	Kinematics	13
2.3.3	Kinetics	14
2.4	Rigid body dynamics and spatial transformations	15
2.4.1	Rigid-body kinetics	15
2.4.2	Spatial algebra	16
2.5	Aerodynamics and hydrodynamics of foils	18
2.5.1	Angle of attack	18
2.5.2	Foil and aerodynamic reference point	19
2.5.3	Foil induced forces and moments	19
2.6	Path following	21
2.6.1	Line-of-sight guidance law	22
2.7	Collision avoidance	23
2.7.1	Velocity obstacle	24
2.7.2	Perception	25
3	Robot and marine craft models	27
3.1	Open chain manipulator model	27
3.1.1	Links and joints	27

x	0.0
---	-----

3.1.2 Joint models	28
3.1.3 Recursive equation of motion	29
3.1.4 Link velocity and acceleration	32
3.1.5 Equation of motion in closed form	34
3.2 Surface vessel model	35
3.2.1 Maneuvering model	35
3.2.2 Hydrodynamic forces	37
3.2.3 Hydrostatic forces	39
4 Modeling	41
4.1 Simulation model	42
4.1.1 Merging the robot and vessel model	42
4.1.2 Vessel model contribution	45
4.1.3 Sail system contribution	46
4.1.4 Joint forces	46
4.1.5 Foil induced forces and gravity	47
4.1.6 Velocity triangle and angle of attack	47
4.1.7 Rudder model	49
4.1.8 Saildrone dynamics overview	49
4.2 Simplified model	50
4.2.1 Assumptions	50
4.2.2 Sail system	52
4.2.3 Simplified rudder and tail model	56
4.2.4 4 DOF Vessel model	56
4.2.5 Rudder contribution	58
4.2.6 Keel contribution	59
4.2.7 Saildrone model	60
4.3 Model comparison and controllability	62
4.3.1 Model comparison	62
4.3.2 Controllability	63
4.4 Model parameters approximation	63
4.4.1 Saildrone inertia	63
4.4.2 Center of gravity and buoyancy	64
4.4.3 Hydrodynamic and hydrostatic parameters	64
4.4.4 Foil coefficients	68
5 Simulation and motion control	73
5.1 Simulator overview	73
5.2 Guidance and control system overview	75
5.2.1 Implementation	76
6 Control and sail propulsion	79
6.1 Reference model	79
6.2 Controllers	80
6.2.1 Tail angle controller	80
6.2.2 Sail AOA controller	82
6.2.3 Heading controller	85

6.3	Sail propulsion	87
6.3.1	Optimal sail AOA	87
6.3.2	Accuracy and dominating dynamics at high sail AOA	91
6.3.3	Sail forces estimate	92
6.4	Course and heading autopilot	92
6.5	Course or heading reference for optimal sail AOA	94
6.6	Controller tuning	95
6.6.1	Tail controller and reference model tuning	95
6.6.2	Sail AOA controller and reference model tuning	96
6.6.3	Heading controller and reference model tuning	99
6.6.4	Tuning summary	102
7	Guidance system	105
7.1	Feasible velocities	105
7.1.1	Convexification of velocity space using tacking and jibing	107
7.1.2	Optimal up- and downwind course	109
7.1.3	Ocean current and feasible velocities	110
7.2	Path following algorithm	114
7.2.1	Course and LOS altering approach	115
7.2.2	LOS based approach	117
7.3	Path manager	119
7.4	Collision avoidance	120
7.5	Guidance system parameters	127
8	Simulations results	129
8.1	Path following comparison	129
8.2	Wind and ocean current compensation when crossing	133
8.2.1	Wind influence	133
8.2.2	Ocean current influence	134
8.3	Jibing and tacking simulation	135
8.4	Improvement by including ocean current	137
8.4.1	Sailing against the wind and ocean current	137
8.4.2	Sailing against the wind with ocean current from the side .	138
8.5	Long mission simulation	141
8.6	Collision avoidance simulation	143
9	Discussion	147
10	Conclusion	151
10.1	Conclusion	151
10.2	Further work	152
A	Path following algorithm pseudo-code	161
B	Aerodynamic data of airfoil	165
C	Properties of homogeneous solids	167

Figures

2.1	Saildrone design	8
2.2	Jibing and tacking	9
2.3	Angle of attack and foil induced forces	18
2.4	Lookahead-based LOS guidance	22
2.5	Collision cone	24
3.1	Links, joints and frames	28
3.2	Generalized forces acting in a link	29
4.1	Saildrone states	42
4.2	Saildrone frames	43
4.3	Generalized forces acting on the saildrone	44
4.4	Velocity triangles	48
4.5	Rudder dynamics block diagram	49
4.6	Saildrone dynamics block diagram	50
4.7	Sail forces	54
4.8	Sail system block diagram	56
4.9	Simplified model block diagram	61
4.10	Open loop comparison of simulation- and simplified model	62
4.11	Lift coefficient approximation	69
4.12	Drag coefficient approximation	70
4.13	Moment coefficient approximation	71
5.1	Simulation example	74
5.2	Guidance and control overview	76
6.1	Sail induced forces	88
6.2	Optimal sail AOA	89
6.3	Force coefficients with respect to reference frame	90
6.4	Sprial test for sail AOA	91
6.5	System response during tacking with heading ψ and course χ as reference for optimal sail AOA	94

6.6	Tail angle step response at $U_w = \{4, 12\}$ m/s	95
6.7	Angles and tail actuator moment under high wind tacking manuver	96
6.8	Sail AOA 15° step response at $U_{rw} = \{4, 12\}$ m/s	97
6.9	Sail AOA under heading change and wind speed $U_w = 16$ m/s . .	98
6.10	Saildrone pose and velocity under heading change and wind speed $U_w = 16$ m/s	99
6.11	Heading dynamics step response under 4 m/s wind at $\beta_w = 90^\circ$.	100
6.12	Heading dynamics step response under 12 m/s wind at $\beta_w = 90^\circ$.	101
6.13	Heading dynamics during upwind tacking under 15 m/s wind . . .	102
6.14	Upwind jib at 5 m/s wind at $\beta_w = 180^\circ$	102
7.1	Steady state analysis for varying wind speed	106
7.2	Generic feasible velocity space diagram (FVD)	107
7.3	Linear combination of velocity vectors	108
7.4	Generic convex velocity space diagram (CVD)	109
7.5	A normalized homogeneous FVD with $v_c = 0$	111
7.6	Velocity vector transformation	111
7.7	CVD including ocean current	113
7.8	Crossing path	115
7.9	LOS based crossing path	118
7.10	Velocity obstacles and free cones	122
7.11	VO when tacking or jibing	123
8.1	Margin to dead zone when using LOS for crossing	130
8.2	Crossing using LOS-based path following	131
8.3	Course and VMG velocity for conducted comparison	132
8.4	Crossing with 8 m/s wind	134
8.5	Crossing with 8 m/s wind and 0.5 m/s ocean current	135
8.6	Upwind tacking and downwind jibing	136
8.7	Upwind and downwind jibing	137
8.8	Keel and rudder AOA during tacking at 12 m/s wind and 1 m/s current	140
8.9	Crossing with and without considering ocean current contribution .	141
8.10	Long mission under charging wind magnitude	142
8.11	States during long mission	142
8.12	Collision avoidance while crossing at 10 m/s wind	143
8.13	Collision avoidance while crossing at 10 m/s wind	144
8.14	Desired path angle during collision avoidance maneuver	145
B.1	NACA-0015 experimental lift coefficient	165
B.2	NACA-0015 experimental drag coefficient	166
B.3	NACA-0015 experimental moment coefficient	166

Tables

2.1	System parameter definitions	11
2.2	Position, velocity, force and moment vector definitions	12
3.1	Joint model definitions	29
4.1	Saildrone dimensions and subsystem weight	67
4.2	Key position vectors	68
5.1	Symbol and plot description	74
6.1	Controller and reference model parameters	103
7.1	Optimal course angle with respect to wind direction at zero current	110
7.2	State and variable definitions algorithm 2	125
7.3	Guidance system key parameters	127
8.1	Course switching and LOS based path planning finish times	132
8.2	Velocities and finish times for 1 km path against the wind and current, without accounting for current.	138
8.3	Velocities and finish times for 1 km path against the wind and current, while accounting for current.	138
8.4	Velocities and finish times for 1 km path against the wind and $\beta_c = 90^\circ$, without accounting for current.	139
8.5	Velocities and finish times for 1 km path against the wind and $\beta_c = 90^\circ$, while accounting for current.	140
A.1	State and variable definitions algorithm 3	161

Acronyms

AC	Aerodynamic Center
AOA	Angle of Attack
BC-MPC	Branching-course MPC
CC	Collision Cone
CG	Center of Gravity
COLAV	Collision avoidance
COLREG	International regulations for avoiding collisions at sea
CVD	Convex velocity diagram
DOF	Degree Of Freedom
EoM	Equations of Motion
FVD	Feasible velocity diagram
GWP	Global Waypoint
LOS	Line-Of-Sight
MPC	Model Predictive Control
NED	North East Down
SAW	Sail And Counterweight
SPP	Sail Pivot Point
USV	Unmanned Surface Vehicle
VMG	Velocity Made Good
VO	Velocity Obstacle
WP	Waypoint

Nomenclature

α	Angle of attack
β	Sideslip angle
β_{crab}	Crab angle
β_c	Direction of ocean current
β_{rw}	Direction of relative velocity of craft w.r.t. ocean current
β_{rw}	Direction of relative velocity of craft w.r.t. wind
β_w	Direction of wind
χ	Course angle
χ_r	Relative course angle
δ_r	Rudder deflection angle
δ_t	Tail deflection angle
η	Generalized position, position and orientation vector
ν	Generalized velocity
ϕ	Roll angle
π_p	Path (tangential) angle
ψ	Heading angle
θ	Pitch angle
ε	Angle of sail w.r.t. vessel heading
ξ	Sail system heading

xx

q State vector of joint variables

T Draft of hull

U Velocity magnitude

Introduction

1.1 Background and motivation

Stretching far back in history, sailing has been an essential part of exploration and transportation [1]. By exploiting the wind, the propulsion method offers long mission duration without the constraints imposed by transitional energy sources. Harvesting energy from the environment rather than carrying fuel for propulsion. These properties, combined with an increasing focus on ocean monitoring and renewable energy transportation, have made sail-propelled craft attractive.

As a result, the application and development of unmanned surface vehicles propelled by sail have seen increasing interest in recent years [2], with fields of applications ranging from seafloor mapping and sea state monitoring to ocean surveillance. These applications are enabled by the combination of modern automatic control, autonomy and the ancient art of sailing, where the craft is required to navigate to the desired destinations without human intervention. Furthermore, these crafts typically combine sail propulsion with solar panels and batteries, enabling crafts to be self-sustainable in terms of energy for actuators, computers and sensor systems.

Despite the possibilities enabled by sail propulsion, the motion control problem becomes non-trivial due to the constraints imposed by the propelling sail, as the craft cannot sail straight against the wind. Instead, the craft has to move in a zig-zag pattern to keep a sufficient angle with respect to the wind direction for the sail to provide propulsion. In this thesis, the general term "crossing" is used to describe this composition of maneuvers. Another important aspect is that the sail-boat relies entirely on a non-zero relative velocity with respect to the surrounding water to be controllable. Without velocity, the rudder-induced forces are zero, and the system does not have an input for steering. These aspects combined impose yet another challenge in sailing; not every maneuver provides propulsion all the

way through. Hence the sailboat velocity will decrease, increasing the risk of losing control over the craft. Meaning alternative, less effective maneuvers need to be applied. Besides the influence of wind, and the fact that no wind also implies no sail propulsion, sailboats are also subjected to the ocean current. Meaning the ocean current affects the motion control problems as well.

For autonomous crafts in general, collision avoidance is a crucial element to enable these to operate in an environment with other objects than themselves. Collision avoidance implies constraining the possible paths or velocities a craft can utilize in order to avoid or significantly reduce the risk of a collision. Due to the limitations of sail propulsion, the dependency on environmental conditions and the required maneuvers to compensate for these effects, the collision avoidance constraints must be significantly more conservative compared to conventional crafts. This results in a small solution space, which might require larger deviations from the original path, further complicating the motion control problem.

1.2 Previous and related work

A motion control system enabling a sailboat to navigate between two destinations autonomously will consist of multiple subsystems working together. Approaches within the main building block of such a motion control system are presented here. A survey of designs, development and deployment of autonomous sailboats can be found in reference [2], where the authors conclude that most of the existing sailboats only support path following and are not entirely autonomous from their perspective, as they lack the capability of collision avoidance and global path planning.

As the induced forces of the sail rely on its attitude with respect to the wind, an essential part of sailing is to maintain a sail angle that provides propulsion and exploits the wind in the best way possible by maximizing the forward propulsion. Optimization of the sail angle is studied in [3], where prior obtained forces data are used to map out the desirable sail angle for any combination of heading and wind direction. In reference [4] an extremum-seeking controller is developed to maximize the forward velocity. Optimizing the sail angle in real-time by measuring the craft velocity and sail angle. Another approach is described in reference [5], where fuzzy logic is applied for the same purpose. With the fuzzy reasoning based on the psychical characteristics of the aerodynamics of the sail. In reference [6] the sail angle is optimized using a cost function in order to bound the roll angle while maximizing the forward force induced by the sail. Where a simplified expression for the sail-induced forces and moments is utilized in the optimization.

Heading or course control is also crucial in sailing. This control problem is similar to conventional crafts, but sway and roll dynamics will be more evident due to the forces introduced by the sail and keel. Various methods have been applied

specifically to sailboats. In reference [7] a nonlinear heading controller is derived by applying integrator backstepping, while [8] utilized \mathcal{L}_1 adaptive control theory. In reference [9] fuzzy logic is applied. Other approaches, such as [10] utilize gain scheduling PI-control, where the gain scheduling is obtained by selecting between a set of pre-tuned controllers. An optimization carries out the selection with respect to the current maneuver.

Multiple methods have been applied to facilitate path following for sailboats. In reference [11] a simple algorithm is proposed, where it changes between course control and potential field guidance depending on the direction of travel relative to the wind. A similar approach is found in [12], using line-of-sight guidance instead of potential field guidance. Another approach is found in [13], where a routing algorithm identifies the optimal heading to maximize the velocity towards the target. A pure line-of-sight guidance approach is used in [3], while [10] optimizes an upwind path by determining the optimal waypoints for course switching. A reactive path planning method is proposed in [14], where a potential field algorithm is applied to drive the craft towards the goal and incorporate the constraints given by the sail and wind. This approach also features collision avoidance as obstacles can be incorporated into the potential.

For collision avoidance, there also exists a few methods that have been applied specifically to sailboats, as the potential field method previously mentioned. In reference [11] interval analysis is used to select between progressing along the path or doing station keeping. Courses towards obstacles are penalized based on distance in [13], resulting in a minimization problem, where only straight courses that provide propulsion are evaluated. Learning methods have also been applied and tested in simulations, like in [15], where reinforcement learning is utilized.

Despite the already existing contributions toward autonomy and automatic control for sailboats, this is still an open field where both the problem structure and the methods applied vary. Within the studied literature, it was observed that little effort has been focused towards the influence of ocean current and how it can be compensated or accounted for. Another interesting finding was that few path following algorithms consider the velocity reduction and potential lack of propulsion during a maneuver, which potentially can result in an uncontrollable system. In addition, despite being a large field in maritime industry and a hot topic in research, most studies considering marine collision avoidance are directed towards conventional propelled crafts. In this thesis a motion control system and simulation framework will be developed with the aim of proving its working concept, but also provide new ideas and contributions addressing the mentioned topics.

Remark: The author of this thesis work on quadrotor collision avoidance with model predictive control in the specialization project, TTK4551 - Engineering Cybernetics. Even though the topics of this thesis is different, some of the theoretical

explanations are reused. This concerns section 2.3.2, 2.3.3 and 2.7.1, these are similar to the specialization project.

1.3 Problem description

The scope of this thesis is to combine ancient technology (sails) with modern technology (automatic control) in order to derive an efficient motion control system for an autonomous sailboat. The task will involve the following steps:

- Derive and establish a simulation model for an appropriate sailboat design, including the effects of currents and winds and the associated fluid-mechanical forces.
- Design a control system.
- Design a path following algorithm that allows the sailboat to sail in any desired direction while subject to the prevailing wind and current.
- Design a collision avoidance algorithm for avoidance of static obstacles like small islands, shallow water, e.t.c.
- Design a motion control system that permits the sailboat to follow a predetermined path autonomously.
- Simulate and evaluate the working concept of the motion control system as a whole. Also showcase other important findings.

1.4 Approach

To enable the sailboat to sail in any direction, avoid collisions and follow a pre-defined path autonomously, a multilayer motion control system is utilized, with a top-level guidance system and lower-level control system.

An optimal sailing path is a combination of the optimal course and the optimal configuration of the sail. The alignment of the sail and the control of the course are eventually separate problems. However, they are connected and need to operate together to provide propulsion in the desired direction. Using this perspective, one can divide the optimality into two sub-problems: the optimal sail configuration given the heading of the craft and the optimal course to progress the fastest to the destination. Hence the control system is divided between control of the craft's heading and the sail's angle of attack.

An offline optimization is used to create a lookup-table for the optimal angle of attack of the sail for the given wind condition and craft heading. The lookup-table combined with the sail angle of attack controller ensures that the optimal angle of attack is maintained without explicit control from the guidance system, which simplifies the guidance system.

The guidance system consists of 3 separate algorithms, path management, path following and collision avoidance. Where the path following algorithm calculate the desired course and decide how to best progress along the desired path. Here ocean current is accounted for in the calculation of the desired course, and alternative maneuvers are selected if lack of controllability is a threat. The collision avoidance algorithm manipulates the desired path if avoidance is required, where obstacles are introduced as velocity vector constraints. The path management is a simple waypoint switching mechanism, changing the waypoint as the craft progresses.

1.5 Contributions

The main contributions of the work presented in this thesis are as follows:

- A sailboat model combining Fossen's robot-inspired marine craft model with the open chain dynamics of a robot manipulator.
- A simplified sailboat model based on Fossen's robot-inspired marine craft model and the contributions from the rudder and keel.
- Heading and sail angle of attack gain-scheduled controllers based on non-dimensional velocity invariant linear systems.
- Sail angle of attack optimization based on approximated models of the foil's lift and drag coefficients.
- A method to introduce the effect of ocean current into the diagram of feasible velocities, allowing selection of courses that are optimal both with respect to wind and ocean current. The increased effective velocity when sailing upwind is demonstrated in simulations.
- A path following algorithm featuring the traditional sailing maneuvers and line-of-sight guidance.
- A simple collision avoidance algorithm where obstacles are represented by constraints in velocity space.
- Simulation studies showing the working concept of the motion control system as a whole.
- Implementation of the simulation framework consisting of the developed models and a simplified perception model, together with the motion control system.
- Animation of the sailboat's response and visualization of poses along the obtained path.

1.6 Limitations

Path planning can be divided into two levels: The global path, where the optimal route from the initial position to the end destination or via points is found considering system limitations and environmental conditions and forecasts. The local

path, concerned with how to progress between the global waypoint when considering the current environmental conditions and the system constraints. This thesis will focus on the latter, where it is assumed that the waypoints are provided by a user.

1.7 Assumptions

The basic assumptions used in this thesis are the following:

- The system is considered fully observable. All values, which in reality may have to be estimated, may be used in calculations.
- The environment is ideal, wind and current are present, but there are no waves and wind gusts.
- The Earth is flat.

1.8 Outline

The remaining of this report is structured as follows. After an introduction to the fundamental theory and notations in chapter 2, the basic modeling framework for robots and marine crafts is presented in chapter 3. In chapter 4, the two models from the previous chapter are merged to create a simulation model. In addition, a simplified model is derived in the same chapter. Further, the developed simulation framework, together with the structure of the motion control system, is presented in chapter 5. The control system is developed through chapter 6, together with an optimization for the sail angle of attack. Chapter 7 introduced the guidance system together with the derived method to exploit ocean current in course selection. In chapter 8, a set of simulation cases are studied, both proving the concept of the motion control system and visualizing the potential performance gain by exploiting ocean current. Finally, the findings are discussed in chapter 9, while concluded remarks and suggested further work is presented in chapter 10.

Chapter 2

Background and essentials

This chapter address relevant definition and topic within sailing, modeling basics, guidance and collision avoidance.

2.1 System description and definitions

The thesis focuses on a saildrone which essentially is an unmanned surface vehicle (USV) that uses a sail for propulsion. The saildrone consists of the following main parts: Hull, keel, rudder, sail, tail-sail and counterweight. The tail-sail is referred to as the tail throughout the thesis. The full drone is shown in figure 2.1. The angle of the sail is controlled using the angle of the tail, while the heading of the saildrone is controlled using the rudder. The joint connecting the hull and the sail is assumed frictionless and features no actuator.

The actuators for the tail and rudder deflection angle are electric and powered by an onboard battery that is charged using a set of solar panels. Drones of this design also feature a backup propulsion system, an electric motor for propulsion during long-term calm wind conditions to avoid grounding or collisions. Only conditions with wind will be evaluated in this thesis, leaving out conditions where the electric motor can be used.

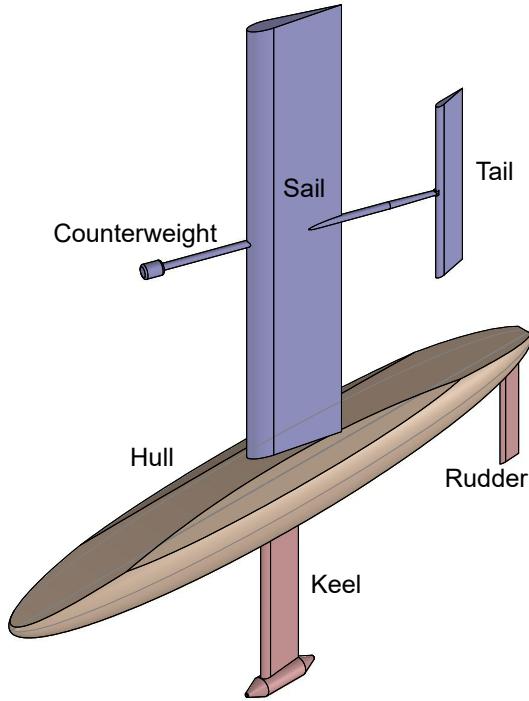


Figure 2.1: Saildrone design

The saildrone consist of the following subsystems referred to through the thesis:

Vessel: Hull together with the keel and rudder.

Sail system: Sail together with the tail and counterweight.

Saildrone: Vessel and sail system.

SAW: Sail and counterweight.

Through the thesis, the connection point between the vessel and the sail system, which is the point the sail pivot around, is referred to as the Sail Pivot Point (SPP).

2.2 Terminology and sailing maneuvers

A craft propelled using sails relies on the lift and the drag induced by the wind in order to create forward propulsion. This also implies that there is a set of course angles against the wind in which the craft can not sail forward, which is often referred to as the dead zone. To travel against the wind, the craft needs to go in a zigzag pattern such that its course relative to the wind is such that the sail can provide propulsion. The zigzagging result in a combination of two courses that give an effective velocity in the desired direction. The zigzagging pattern is also used when sailing along the direction of the wind, as the effective velocity obtained can be higher than when going straight.

Another important property of sailing is that the craft relies on relative velocity to be controllable. Without wind, there is no propulsion. However, without relative velocity with respect to the surrounding water, the rudder does not provide any lift or drag to produce a turning moment.

To perform the zigzagging or crossing as it will be defined, two types of maneuvers are used, tacking and jibing. Their definition vary, however in this thesis they are defined as:

- **Tacking** is when the bow (front) of the craft is steered through the direction against the wind, as illustrated to the left in figure 2.2. Notice that propulsion is lost during a segment of this maneuver as the bow points against the wind.
- **Jibing** is when the craft is steered such that the stern (rear of the craft) passes against the wind direction, as shown by the two illustrations to the right in figure 2.2. Here there are two different cases, downwind jibing and upwind jibing. Where upwind jibing is used when a tacking maneuver might cause the craft to lose control, as the upwind jibing always provides forward propulsion as opposed to a tacking maneuver.
- **Crossing** is when sailing in a zigzag motion against or along with the wind, combining straight line segments with tacking or jibing. This is not a traditional term used in sailing, however with respect to motion control and guidance, it is useful.

Notice that upwind jibing might also be required for any course change that originally demands the bow to cross against the wind.

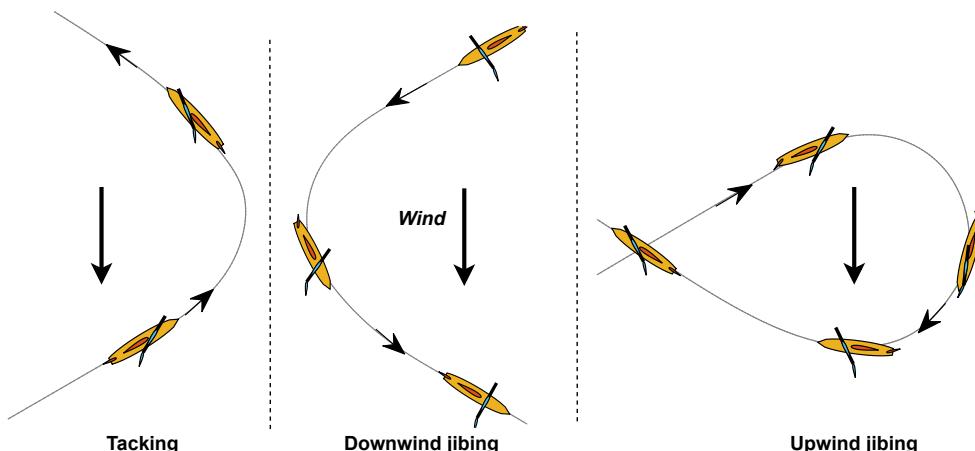


Figure 2.2: Jibing and tacking

To describe the motion of the saildrone the following basic definitions are utilized:

- **Heading** is the angle between north and the x-axis of the saildrone
- **Course** is the angle between north and the saildrone's velocity vector.
- **Surge** is the motion along the saildrone's x-axis
- **Sway** is the motion along the saildrone's y-axis
- **Heave** is the motion along the saildrone's z-axis
- **VMG** velocity is the velocity of the craft projected into the path it follows.

2.3 Modeling preliminaries and definitions

In this section, the applied notation and reference frames are defined, together with the kinematic and kinetic equations.

2.3.1 Reference frames and definitions

For a USV operating in a local area without any motion constraints, 6 degrees of freedom (DOF), the position and orientation of the body frame relative to the inertial frame can be described as in equation 2.1. Where the orientation is parameterized using the Euler angles. While the linear and angular velocities in the body fixed frame are defined as equation 2.2.

$$\eta = [x^n \ y^n \ z^n \ \phi \ \theta \ \psi]^T \quad (2.1)$$

$$\nu = [u \ v \ w \ p \ q \ r]^T \quad (2.2)$$

The equations of motion are derived using 6DOF formulation due to the convenience of spatial algebra presented in section 2.4.2. Reduced models with fewer degrees of freedom are also used throughout the thesis.

The frames are defined using the North-East-Down (NED) convention. In addition, the Earth's rotation is neglected and the local earth-fixed NED frame is considered inertial. For simplicity, the Earth is assumed to be flat.

For the derivation of the system dynamics and further in this thesis the definitions in table 2.2 and 2.1 will be used. To describe the different frames, forces and moments the notation $\{_$ is adapted along with the following coordinate definitions. Recursive equations with respect to frames are also used, hence the relevant frames are both represented by numbers and letters.

Reference frames:

- $\{n\} = \{0\}$: NED frame (assumed to be inertial).
- $\{b\} = \{1\}$: Body frame.
- $\{s\} = \{2\}$: Sail frame.
- $\{t\} = \{3\}$: Tail frame.

- {wf}: Wind flow frame.
- {cf}: Current flow frame.
- {i}: Link reference frame
- {i-1}: Link predecessor reference frame.
- {i+1}: Link successor reference frame.
- {p}: Path reference frame.

Reference points:

- CO: Coordinate origin of the body frame {b}.
- CG: Center of gravity.
- CF: Center of flotation, the centroid of the waterplane area A_{wp} .
- CB: Center of buoyancy, the volume centroid of the displaced water.
- CP: Center of at generic point.
- AC: Aerodynamic center.

Subscripts:

- $_{rw}$: Subscript for relative velocity with respect to wind.
- $_{rc}$: Subscript for relative velocity with respect to ocean current.

Table 2.1: System parameter definitions

Description	Notation
Position vector from a to b w.r.t. a	$r_{ab}^a = \begin{bmatrix} r_{ab,x}^a \\ r_{ab,y}^a \\ r_{ab,z}^a \end{bmatrix}$
Inertia of the rigid body in b w.r.t. a	$I_b^a = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{yx} & I_{yy} & -I_{yz} \\ -I_{zx} & -I_{zy} & I_{zz} \end{bmatrix}$

The generalized velocity and generalized force are written as ν and τ , respectively. Implicitly meaning ν_{nb}^b and τ_b^b . For generalized vectors in other frames, the full expression will often be used. However, for the iterative equations, the motion and generalized force vectors are always described with respect to the link frame and the full notation becomes overwhelming. Hence the reduced notation is used $\nu_{ni}^i = \nu_i$. In general, vector components for forces and moments are expressed with upper case letters as shown in table 2.2.

Table 2.2: Position, velocity, force and moment vector definitions

Description	Notation
Position of CO in inertial/NED frame	$p_{nb}^n = \begin{bmatrix} x^n \\ y^n \\ z^n \end{bmatrix}$
Linear velocity in body frame	$v_{nb}^b = \begin{bmatrix} u \\ v \\ w \end{bmatrix}$
Attitude parameterized by Euler angles, from {b} to {n}	$\Theta_{nb} = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix}$
Angular velocity of {b} with respect to {n} expressed in {b}	$\omega_{nb}^b = \begin{bmatrix} p \\ q \\ r \end{bmatrix}$
Forces expressed in body frame	$f_b^b = \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix}$
Moment about CO expressed in body frame	$m_b^b = \begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix}$
Generalized force vector with respect to CO in {b}	$\tau = \begin{bmatrix} f_b^b \\ m_b^b \end{bmatrix}$
Generalized velocity vector with respect to CO in {b}	$v = \begin{bmatrix} v_{nb}^b \\ \omega_{nb}^b \end{bmatrix}$
Vector of joint force inputs	\mathcal{T}

In robot literature, it is common to use q for the joint states and \dot{q} for the angular velocity with respect to the body y-axis. Hence these are separated by using "Roman" q for the joint states and "Italic" \dot{q} for the angular velocity.

m denote both mass, motion vectors and moment; however, it should be clear from the context which of these are used at the specific place.

2.3.2 Kinematics

The transformation between reference frames is carried out using rotation matrices parameterized with Euler angles. All rotations can be obtained by combining three principal rotations. For this thesis, the zyx-convention is used, representing rotations by three principal rotations around the z, y and x axes.

The principal rotations are represented by:

$$R_{x,\phi} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\phi & -s\phi \\ 0 & s\phi & c\phi \end{bmatrix}, R_{y,\theta} = \begin{bmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{bmatrix}, R_{z,\psi} = \begin{bmatrix} c\psi & -s\psi & 0 \\ s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.3)$$

Where s and c represent cos and sin of the angles.

The rotations can be combined to form the transformation from one frame to another. The transformation from the body frame to the NED frame becomes equation 2.4. By the properties of the rotation matrices, the transpose rotation matrix is the inverse operation, shown in equation 2.5.

$$R_b^n = R_{z,\psi} R_{y,\theta} R_{x,\phi} \quad (2.4)$$

$$R_i^b = R_b^{i^T} \quad (2.5)$$

Considering the transformation between the Euler angle rates $\dot{\Theta}_{nb}$ and the angular velocity around the body axes ω_{nb}^b , the relation is more complex as the Euler angles describe rotations between intermediate frames. The transformation can be written as:

$$\dot{\Theta}_{nb} = T(\Theta_{nb}) \omega_{nb}^b \quad (2.6)$$

Where the transformation $T(\Theta_{nb})$ and its inverse are:

$$T(\Theta_{nb}) = \begin{bmatrix} 1 & s\phi t\theta & c\phi t\theta \\ 0 & c\phi & -s\phi \\ 0 & \frac{s\phi}{c\theta} & \frac{c\phi}{c\theta} \end{bmatrix}, T(\Theta_{nb})^{-1} = \begin{bmatrix} 1 & 0 & -s\theta \\ 0 & c\phi & c\theta s\phi \\ 0 & -s\phi & c\theta c\phi \end{bmatrix} \quad (2.7)$$

where s , c and t represent cos, sin and tan of the angles.

The derivation of the transformations can be found in reference [16]. Equation 2.7 also show that the transform is singular for a specific attitude. Which is one of the main disadvantages of the Euler angles.

Summarizing the transformations gives the 6DOF kinematic equations, which can be expressed as equation 2.8. Note that the derived transformation is valid for all

subsystems. Hence it can be formulated for any frame. The body and NED frame are only used as an example.

$$\underbrace{\begin{bmatrix} \dot{p}_{nb}^n \\ \dot{\Theta}_{nb} \end{bmatrix}}_{\dot{\eta}} = \underbrace{\begin{bmatrix} R(\Theta_{nb}) & 0_{3 \times 3} \\ 0_{3 \times 3} & T(\Theta_{nb}) \end{bmatrix}}_{J_\Theta(\eta)} \underbrace{\begin{bmatrix} v_{nb}^b \\ \omega_{nb}^b \end{bmatrix}}_v \quad (2.8)$$

2.3.3 Kinetics

The equation of motion can be derived using the Newton-Euler formulation, which is based on Newton's second law and Euler's first and second axiom considering the conservation of linear and angular momentum. As mentioned earlier, the NED frame will be considered the inertial frame, and each part of the saildrone is assumed to be rigid.

As Newton's law applies in the inertial frame, the rotational motion has to be considered when time differentiating a vector \vec{r} in a rotating reference frame. This is due to the fact that the vector change relative to the inertial frame as a result of two effects. Firstly, change to the vector viewed from within the rotating frame. Secondly, the vector change due to the change of orientation of the rotating frame relative to the inertial. These can be formulated:

$$\frac{^i d}{dt} \vec{r} = \frac{^b d}{dt} \vec{r} + \omega_{ib} \times \vec{r} \quad (2.9)$$

Where $\frac{^i d}{dt}$ is the time differentiation with respect to the inertial frame (NED), and $\frac{^b d}{dt}$ is with respect to the rotating frame (e.g. body frame).

This relation, together with the definitions in table 2.2 and 2.1, can be utilized to define the translational and rotational equation of motion around CG within $\{b\}$. The translational motion of CG can be derived as follows:

$$\vec{f}_g = \frac{^n d}{dt} (m v_{ng}) \quad (2.10)$$

$$f_g^b = m (\dot{v}_{ng}^b + \omega_{nb}^b \times v_{ng}^b) \quad (2.11)$$

The rotational motion about CG can be derived:

$$\vec{m}_g = \frac{^n d}{dt} (I_g \omega_{nb}) \quad (2.12)$$

$$m_g^b = I_g^b \dot{\omega}_{nb}^b + \omega_{nb}^b \times (I_g^b \omega_{nb}^b) \quad (2.13)$$

Where I_g and v_{ng}^b is the inertia and the velocity with respect to the CG in $\{b\}$. A more exhaustive derivation can be found in [16–18].

2.4 Rigid body dynamics and spatial transformations

In this section, the rigid body dynamics is presented along with the transformations applied for modeling the multilink saildrone. A compact 6-dimensional representation is used, combining the translational and rotational aspects of rigid-body motion and forces. The representation and the associated mathematics are often referred to as spatial vectors and spatial vector algebra in robot literature [19]. The spatial representation allows the equation of motion to be written in matrix form. The following section is based on reference [19], [20] and [16]. Notice that some of these references use notation with the rotational dynamics first, while in this thesis, the translational dynamics is placed first.

2.4.1 Rigid-body kinetics

Utilizing the generalized force vector from table 2.2 together with equation 2.11 and 2.13 the rigid-body equations of motion (EoM) with respect to CG can be written on matrix form as equation 2.14 [16].

$$\underbrace{\begin{bmatrix} mI_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & I_g^b \end{bmatrix}}_{M_{RB}^{CG}} \begin{bmatrix} \dot{v}_{ng}^b \\ \dot{\omega}_{nb}^b \end{bmatrix} = \underbrace{\begin{bmatrix} mS(\omega_{nb}^b) & 0_{3 \times 3} \\ 0_{3 \times 3} & -S(I_g^b \omega_{nb}^b) \end{bmatrix}}_{C_{RB}^{CG}} \begin{bmatrix} v_{ng}^b \\ \omega_{nb}^b \end{bmatrix} = \begin{bmatrix} f_g^b \\ m_g^b \end{bmatrix} \quad (2.14)$$

Where M_{RB}^{CG} is the rigid-body system inertia matrix with respect to CG. While C_{RB}^{CG} is the Coriolis-centripetal matrix with respect to CG. $S(x)$ is the cross-product operator that maps $x \in \mathbb{R}^3$ into a skew symmetric matrix.

It is convenient to express the equations of motion within other reference point than the CG. This can be achieved by using the system transformation matrix presented in section 2.4.2. Transforming equation 2.14 into the CO give equation 2.15.

$$M_{RB} \underbrace{\begin{bmatrix} \dot{v}_{ng}^b \\ \dot{\omega}_{nb}^b \end{bmatrix}}_{\dot{v}} + C_{RB}(\nu) \underbrace{\begin{bmatrix} v_{ng}^b \\ \omega_{nb}^b \end{bmatrix}}_v = \underbrace{\begin{bmatrix} f_g^b \\ m_g^b \end{bmatrix}}_{\tau} \quad (2.15a)$$

$$M_{RB} = H^T(r_{bg}^b) M_{RB}^{CG} H(r_{bg}^b) \quad (2.15b)$$

$$C_{RB}(\nu) = H^T(r_{bg}^b) C_{RB}^{CG} H(r_{bg}^b) \quad (2.15c)$$

$$\tau = H^T(r_{bg}^b) \begin{bmatrix} f_g^b \\ m_g^b \end{bmatrix} \quad (2.15d)$$

The transformed Coriolis-centripetal matrix at CO can be written as equation 2.15c. Notice that C_{RB} will be independent of the linear velocity, which is a useful property exploited in section 3.2.

2.4.2 Spatial algebra

Within spatial algebra, motion and force vectors are subjected to different transformation rules due to the difference in how force and motion are related when observed from different frames and points. Despite the difference, the transformations are closely related as described by the system and coordinate matrix transformations presented in the coming section.

The structure of these transformations can be revealed by writing out the generalized velocity and force vector relating the *CP* and *CO*, as shown in equation 2.16 and 2.17. The linear motion is caused by both linear and angular motion and will hence vary over the body. In contrast, the angular motion is constant throughout the entire rigid body. For generalized forces, the sum of forces is constant throughout the entire rigid body. The moment is caused by both pure moment and the moments created by the applied forces and their respective point of application.

$$\begin{bmatrix} v_{np}^b \\ \omega_{np}^b \end{bmatrix} = \begin{bmatrix} v_{nb}^b + \omega_{nb}^b \times r_{bp}^b \\ \omega_{nb}^b \end{bmatrix} \quad (2.16)$$

$$\begin{bmatrix} f_b^b \\ m_b^b \end{bmatrix} = \begin{bmatrix} f_p^b \\ r_{bp}^b \times f_p^b + m_p^b \end{bmatrix} \quad (2.17)$$

System transformation matrix

The system transformation matrix H transforms generalized velocities, accelerations and forces between two points within the same rigid body. The transformation between a reference point *CP* and *CO* is shown in equation 2.18. The transform can be derived directly from equation 2.16 and 2.17, which is done in details in reference [16] (appendix C).

$$H(r_{bp}^b) = \begin{bmatrix} I_{3 \times 3} & S^T(r_{bp}^b) \\ 0_{3 \times 3} & I_{3 \times 3} \end{bmatrix}, \quad H^{-1}(r_{bp}^b) = \begin{bmatrix} I_{3 \times 3} & S(r_{bp}^b) \\ 0_{3 \times 3} & I_{3 \times 3} \end{bmatrix} \quad (2.18)$$

It is important to notice that the transform is applied differently to forces and motion vectors as seen in equation 2.19 and 2.20.

$$v_{nb}^b = H^{-1}(r_{bp}^b)v_{np}^b \quad (2.19)$$

$$\tau_b^b = H^T(r_{bp}^b)\tau_p^b \quad (2.20)$$

Coordinate transformation matrix

The coordinate transformation matrix K transforms motions and force vectors between different points and reference frames within a multi-rigid-body system.

The transformation can be viewed as a generalization of the system transformation matrix, and they are related by the rotation matrix.

The motion transformation between to generic frames a and b :

$$K_b^a(r_{ab}^a, R_b^a) = H^{-1}(r_{ab}^a) \begin{bmatrix} R_b^a & 0_{3 \times 3} \\ 0_{3 \times 3} & R_b^a \end{bmatrix} = \begin{bmatrix} R_b^a & S(r_{ab}^a)R_b^a \\ 0_{3 \times 3} & R_b^a \end{bmatrix} \quad (2.21a)$$

$$K_b^a(r_{ba}^b, R_b^a) = \begin{bmatrix} R_b^a & -R_b^a S(r_{ba}^b) \\ 0_{3 \times 3} & R_b^a \end{bmatrix} \quad (2.21b)$$

The coordinate transformation can be formulated differently based on the applied position vector r and rotation matrix R as seen from equation 2.21. However, all formulations are equal, hence only a specific set of these equations are presented and the transformation argument is left out.

Denoting the motion transforms with K and the force transforms with $\overset{\tau}{K}$, the transformations can be summarized:

$$K_b^a = \begin{bmatrix} R_b^a & -R_b^a S(r_{ba}^b) \\ 0_{3 \times 3} & R_b^a \end{bmatrix}, \quad K_a^b = \begin{bmatrix} R_a^b & S(r_{ba}^b)R_a^b \\ 0_{3 \times 3} & R_a^b \end{bmatrix} \quad (2.22)$$

$$\overset{\tau}{K}_b^a = \begin{bmatrix} R_b^a & 0_{3 \times 3} \\ -R_b^a S(r_{ba}^b) & R_b^a \end{bmatrix}, \quad \overset{\tau}{K}_a^b = \begin{bmatrix} R_a^b & 0_{3 \times 3} \\ S(r_{ba}^b)R_a^b & R_a^b \end{bmatrix} \quad (2.23)$$

Where the motion and force transformations are related by:

$$\overset{\tau}{K}_a^b = (K_a^b)^{-T} = (K_b^a)^T \quad (2.24)$$

For a detailed derivations see reference [19], [20] and [21].

Similar to the rotation matrix, the transform through a set of frames is produced by a sequence of transformations:

$$K_1^3 = K_2^3 K_1^2 \quad (2.25)$$

Spatial cross product

There exist two different cross products for spatial vectors. The first cross product takes two motion vectors as an argument and results in a new motion vector as seen in equation 2.27. While the second crosses a motion vector with a generalized force vector, resulting in a new generalized force vector as seen in equation 2.28. In addition, similar to the cross product operator $S(x)$ with $x \in \mathbb{R}^3$, a spatial cross product operator can be defined $S^s(x)$ with $x \in \mathbb{R}^6$.

To describe a general motion vector the notation $\underline{m} = [\underline{m}_L \quad \underline{m}_A]^T$ is used, where \underline{m}_L is the linear motion and \underline{m}_A is the angular motion. Applying the formulations found in reference [20], the spatial cross product operator can be written:

$$S^s(\underline{m}) = \begin{bmatrix} S(\underline{m}_A) & S(\underline{m}_L) \\ 0_{3 \times 3} & S(\underline{m}_A) \end{bmatrix} \quad (2.26)$$

Using the operator, the spatial cross product can be written:

$$\underline{m}_1 \times^s \underline{m}_2 = S^s(\underline{m}_1) \underline{m}_2 \quad (2.27)$$

$$\underline{m} \times^s \tau = -(S^s(\underline{m}))^T \tau \quad (2.28)$$

Notice that $S^s(\underline{m})$ maps motion to motion vectors, while $(S^s(\underline{m}))^T$ maps force vectors to force vectors [20].

2.5 Aerodynamics and hydrodynamics of foils

In this section, the forces and moments induced by a foil moving through a fluid are presented. In addition to the reference point where these forces and moments act. The equations and description are based on reference [18] and the experimental results in [22].

Notice that the reference point for the forces and moments is described as the aerodynamic center (AC), but it is general for all foils surrounded by a fluid.

2.5.1 Angle of attack

Angle of attack (AOA) denoted α , is the angle of the relative velocity vector (with magnitude U_r) with respect to the foil's chord line. Here the relative velocity is the velocity of the foil's aerodynamic center with respect to the surrounding fluid.

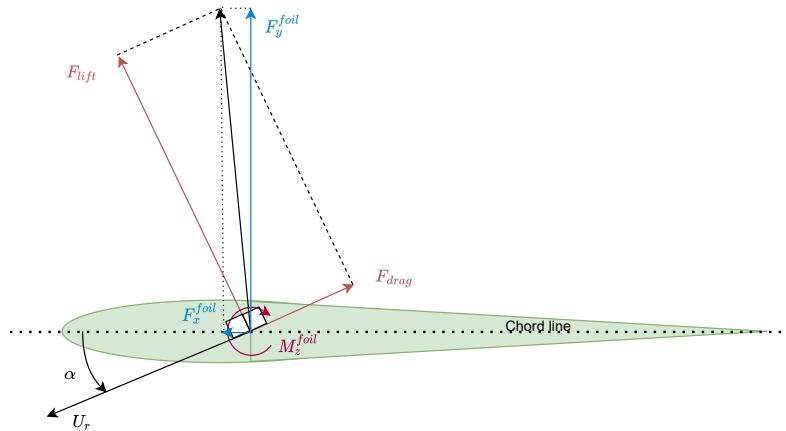


Figure 2.3: Angle of attack and foil induced forces

As illustrated in figure 2.3 the induced lift force acts perpendicular to the relative velocity, while the induced drag act in the opposite direction of the relative velocity.

2.5.2 Foil and aerodynamic reference point

To represent the total lift and drag induced by a foil, a common point where the forces apply is needed. The center of pressure on the foil can be used; however, this point move as pressure varies due to the change of angle of attack. Instead, the aerodynamic center (AC) can be used, which is the point on the foil where the aerodynamic moment remains constant and independent of the angle of attack for moderate AOA. This point can typically be found at the quarter-chord point [23][24]. The chord is defined as the length of the straight line from the tip to the tail of a foil. For a symmetric foil, the moment at this point will be zero for moderate AOA. At low speeds (in the supersonic speed range), most of the aerodynamic forces are induced by the AOA. This, together with the constant moment at AC, makes it a desirable aerodynamic reference point, often used in stability and control studies [24].

In this section, the aerodynamic center is described, but the behavior is similar for all foils operating in a fluid, even though the magnitude will differ. Hence the location of the aerodynamic center will also be valid for a keel or rudder of the same design. Through the thesis, it is referred to AC, meaning the general center for any foil regardless of the surrounding fluid.

2.5.3 Foil induced forces and moments

The forces and moment illustrated in figure 2.3 can be expressed using the following equation [16][18].

$$F_{drag} = \frac{1}{2} \rho U_r^2 S C_D(\alpha) \quad (2.29)$$

$$F_{lift} = \frac{1}{2} \rho U_r^2 S C_L(\alpha) \quad (2.30)$$

$$M_{foil} = \frac{1}{2} \rho U_r^2 S c_m C_m(\alpha) \quad (2.31)$$

Where:

ρ : Density of surrounding fluid.

U_r : Magnitude of relative velocity.

S : Area of the foil.

c_m : The mean chord length of the foil.

$C_L(\alpha)$: Lift coefficient.

$C_D(\alpha)$: Drag coefficient.

$C_m(\alpha)$: Moment coefficient.

These coefficients are highly nonlinear, however for small angle of attack the flow over the foil stays laminar and attached, creating a narrow operation region with coefficients approximately linear with respect to AOA.

Foil induced lift

For small AOA a linear lift model can be used:

$$C_L(\alpha) = C_{L_0} + C_{L_\alpha} \alpha \quad (2.32)$$

$$C_{L_\alpha} = \frac{\pi AR}{1 + \sqrt{1 + (AR/2)^2}} \quad (2.33)$$

Where C_{L_0} will be zero for a symmetric foil. The aspect ratio of the foil is defined as $AR = h^2/S$ where h is the wingspan or foil heights for the saildrone.

Further in this thesis, the following relation is used, which is valid when $C_{L_0} = 0$ and AOA is small:

$$k_{CL} := \frac{1}{2} \rho S C_{L_\alpha} \quad (2.34)$$

An important aspect of lift-inducing foils is the stall angle, which is the angle at which the flow starts to separate from the foil. Resulting in a dramatic drop in lift, which the linear lift model does not reflect.

In order to create a more accurate model accounting for the stall angle, one can assume that the foil behaves approximately like a flat plate when passing the stall angle. Then a nonlinear lift model can be created by weighting the linear and the flat plate model as a function of the AOA [18].

The flat plate model $C_{L_{fp}}(\alpha)$ and the weighting function $\sigma_L(\alpha)$ can be formulated:

$$C_{L_{fp}}(\alpha) = 2\text{sign}(\alpha)\sin^2(\alpha)\cos(\alpha) \quad (2.35)$$

$$\sigma_L(\alpha) = \frac{1 + e^{-J(\alpha - \alpha_0)} + e^{J(\alpha + \alpha_0)}}{(1 + e^{-J(\alpha - \alpha_0)})(1 + e^{J(\alpha + \alpha_0)})} \quad (2.36)$$

Where α_0 describes the cutoff angle for the blending function, and J describes the transition rate between the linear and flat plate model.

The resulting nonlinear lift model can then be formulated:

$$C_L(\alpha) = (1 - \sigma_L(\alpha))(C_{L_0} + C_{L_\alpha} \alpha) + \sigma_L(\alpha)C_{L_{fp}} \quad (2.37)$$

Drag and moment

Linear models for the moment and drag can be obtained by linearization. The moment coefficient of the symmetric foil will remain close to zero for moderate angles [22], hence it is often assumed to be zero. At the same time, a linear drag model will not be symmetric around zero AOA, which causes difficulties when both sides of the foil will be used and negative AOA is common.

The drag coefficient can instead be approximated by a nonlinear function combining the parasitic drag and the induced drag of the angle of attack. Where the parasitic drag, caused by effects like the shear stress of the fluid flowing over the foil, will be independent of the AOA. While the induced drag is proportional to the square of the lift force.

$$C_D(\alpha) = C_{D_p} + \frac{(C_{L_0} + C_{L_a}\alpha)^2}{\pi e_{os} AR} \quad (2.38)$$

Where e_{os} is Oswald efficiency factor, which typically is between 0.8 and 1.0 [25].

Notice that the drag model and zero moment coefficient approximation are only valid for moderate angle of attack.

Flow-axes rotation matrix

The foil-induced forces described by lift and drag are defined in the flow frame, which is a frame with origin at the foil's AC and whose x-axis always point along the relative velocity vector shown in figure 2.3, while the z-axis point along the foil. To rotate the forces from the flow frame to the foil frame where the x-axis is aligned with the chord line, the following formulation can be used:

$$\begin{bmatrix} F_x^{foil} \\ F_y^{foil} \\ M_z^{foil} \end{bmatrix} = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -F_{drag} \\ -F_{lift} \\ -M_{foil} \end{bmatrix} \quad (2.39)$$

2.6 Path following

In this section, path following is distinguished from trajectory tracking. In addition, a line-of-sight guidance law is presented.

Trajectory tracking and path following are often interchanged terms, but these are essentially different. In trajectory tracking, the system is controlled towards a reference point that progresses along a trajectory as time evolves, which means that position along the track is related to time. Path following, on the other hand,

is not concerned with progression along the path, but the objective is only to converge to and follow the path. This means that method is independent of time and the target position is eventually reached by following the path.

The saildrone's feasible velocities and maneuverability greatly rely on the environmental parameters wind and ocean current. Due to these system constraints, a guidance system using path following algorithms is better suited than trajectory tracking. The material presented in this section is based on reference [16, 26].

2.6.1 Line-of-sight guidance law

Line-of-sight (LOS) guidance is a path following algorithm that uses the line between two reference points or waypoints to create a reference path. Where the path is followed by manipulating the desired course in order to minimize the cross-track error y_e^p shown in figure 2.4 while progressing along the path. Where the cross-track error is simply the y-position relative to the path reference frame $\{p\}$. In look-ahead based LOS, the balance between progression along the path and convergence towards the path is adjusted by the look-ahead distance Δ , shown in figure 2.4.

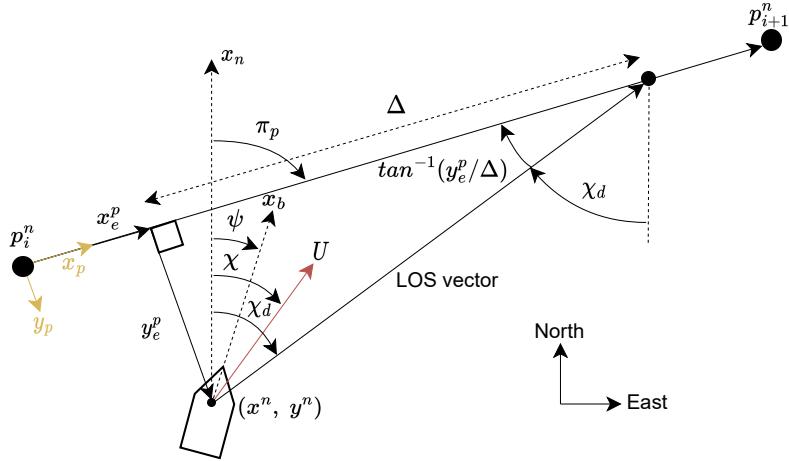


Figure 2.4: Lookahead-based LOS guidance

Figure 2.4 illustrate the principle of lookahead-based LOS guidance where χ_d is desired course, π_p is the path tangential angle and y_e^p is three cross-track error. While Δ is the lookahead distance along the path. The reference points p_i^n and p_{i+1}^n are typically waypoints and the path frame stays attached to the first reference point.

The path tangential angle is given by equation 2.40, where $p_{i+1}^n = [x_{i+1}^n, y_{i+1}^n]^T$. While the along-track and cross-track error are found using equation 2.41.

$$\pi_p = \text{atan}2(y_{i+1}^n - y^n, x_{i+1}^n - x^n) \quad (2.40)$$

$$\begin{bmatrix} x_e^p \\ y_e^p \end{bmatrix} = R_p^n(\pi_p)^T \left(\underbrace{\begin{bmatrix} x^n \\ y^n \end{bmatrix}}_{p_{nb}^n} - \underbrace{\begin{bmatrix} x_i^n \\ y_i^n \end{bmatrix}}_{p_i^n} \right) \quad (2.41)$$

Where

$$R_p^n(\pi_p) = \begin{bmatrix} \cos(\pi_p) & -\sin(\pi_p) \\ \sin(\pi_p) & \cos(\pi_p) \end{bmatrix} \quad (2.42)$$

The desired course is given by equation 2.43, where the look-ahead distance $\Delta > 0$ is a tuning variable that dictates how aggressive the craft will approach the path. With a short look-ahead distance (small value), the LOS vector in figure 2.4 will point more directly into the path, and the cross-track error will be reduced faster. A higher value will give a course angle closer to π_p , and the cross-track error reduction is slower. However, progression along the path is higher. For large ships, a look-ahead distance of two times the length of the ship is often used [26]. Notice that the smaller the cross-track error, the closer the desired course will be to the path tangential angle.

$$\chi_d = \pi_p - \tan^{-1} \left(\frac{y_e^p}{\Delta} \right) \quad (2.43)$$

2.7 Collision avoidance

In this section, the basic idea of collision avoidance is presented. The section also describes the theory of obstacles incorporated in velocity space rather than as positional constraints. The discussion on velocity space obstacles is based on reference [27].

Collision avoidance (COLAV) is the field of handling known or unknown, static or dynamic obstacles by selecting a path that either prevents or minimizes the risk of collision. This is a large field of study, where solutions are often tailored to the specific systems as the collision avoidance problem depends on the system dynamics, system constraints and the space in which it operates. This means COLAV consists of subproblems within situation interpretation and motion planning/control. The complexity of the methods used in COLAV varies, from optimization-based motion planning using model predictive control (MPC) [28] to station-keeping while a dynamic obstacle is blocking the planned path [11].

Within a guidance system, collision avoidance often represents a subsystem that deals with situations where the craft needs to deviate from its original path in order to avoid collisions. Despite the use of different methods to construct these systems, they require robustness as they are safety critical.

2.7.1 Velocity obstacle

Velocity obstacles (VO) are obstacles introduced in velocity space, representing the set of velocities that will result in a collision between the craft and the obstacle. Avoidance maneuvers can hence be created by selecting velocities outside the velocity obstacles. The formulation also makes the constraints imposed by the obstacles independent of the complexity of the contour of the obstacle [29].

The intention of the velocity obstacle is to simplify the motion planning problem by solving it in the velocity space rather than by simultaneously solving the path planning- and velocity planning- problems. The method is a first-order linear approximation of the velocities that will lead to a collision at some future time [27].

Considering a craft and an obstacle modeled as two separate circular objects in 2D, labeled A_0 and B_0 respectively. The problem formulation can be simplified by considering A_0 as a point A , while enlarging/expanding B_0 with the radius r of A_0 to create object B . The craft and the object are associated with a position and a velocity vector v_A and v_B attached to their respective center.

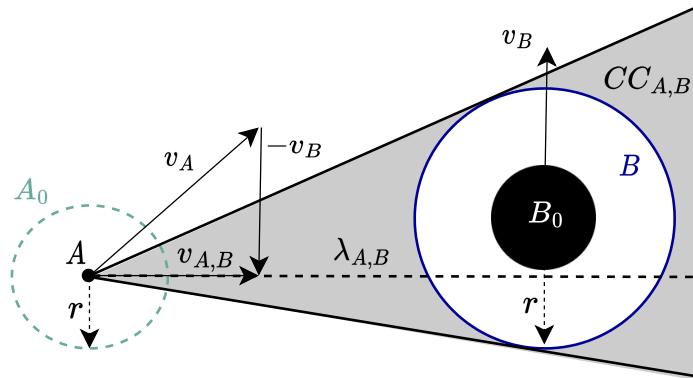


Figure 2.5: Collision cone

The collision cone $CC_{A,B}$ can now be defined as the set of all relative velocities between A and B that will lead to a collision in some future time [29]. Formally this can be written as:

$$CC_{A,B} = \{v_{A,B} \mid \lambda_{A,B} \cap B \neq \emptyset\} \quad (2.44)$$

Where $\lambda_{A,B}$ is the line illustrated in figure 2.5 that coincide with the relative velocity vector $v_{A,B} = v_A - v_B$. In figure 2.5 the gray region illustrate equation 2.44. The VO is defined as all craft velocities v_A that will lead to a collision in some future time. The VO is obtained by transforming the CC from the relative velocity space into the perspective of the craft, by translating the CC by v_B . Formally this can be done using the Minkowski vector sum operator \oplus , where the operator adds each vector in the first set to each vector in the second set. Notice that when $v_B = 0$, implying a static environment, the collision cone $CC_{A,B}$ equal the velocity obstacle $VO_{A,B}$

$$VO_{A,B} = CC_{A,B} \oplus v_B \quad (2.45)$$

2.7.2 Perception

To interpret the environment, a binary occupancy map is used, which is a simplistic representation of the environment. Here a discrete 2D grid makes up cells that are either occupied or unoccupied, implying there are no uncertainties. When obtaining a measurement from the grid, all cells along a finite set of lines originating from the saildrone's position are checked. The measurements are given as intersection points between the lines and the first occupied cell they encounter. The lines themselves are also of finite length, creating a detection range. This simple interpretation is used for collision avoidance, where the obstacles can represent shallow water, island, crafts, e.t.c.

Robot and marine craft models

In this chapter, the open chain manipulator model is presented along with the robot-inspired model for marine crafts. Both are presented using all 6 degrees of freedom. The open chain formulation is a generic model that can describe the dynamics of a chain of connected rigid bodies.

3.1 Open chain manipulator model

In this section, a framework for robot modeling is presented where spatial vectors are utilized. A general model structure is derived using a recursive formulation, allowing each individual link of a multi-rigid body system to be modeled separately. In addition, the relevant joint models are presented.

The derivation of the model of this system dynamics is based on reference [21] combined with [20] and [19]. Here every link frame is placed at the joint connecting link i to the predecessor link $i - 1$, for rotational joints.

3.1.1 Links and joints

To describe the interaction between links, the following definitions are applied. The successor link is the next link in the chain denoted by link $i + 1$. The predecessor link is the previous link in the chain denoted by link $i - 1$. The base frame is the inertial reference frame in the system.

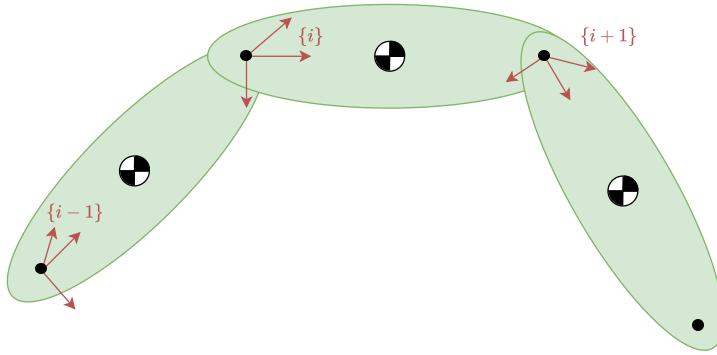


Figure 3.1: Links, joints and frames

Each link i is defined with its own body frame $\{i\}$, where the frame is placed at the joint connecting the link to its predecessor or base frame. In this way, the joints can also be related to a particular link, and the joint number is inherited from the link number. The base frame is denoted $\{0\}$ and is assumed to be inertial. $\{0\}$ has no predecessor, hence it has no associated joint. This means the joint number $\in [1, n]$ while the frame number $\in [0, n]$, where n is the number of links in the system. Using this relation, i will be used to denote both the link, joint and frame number.

3.1.2 Joint models

In a chain or tree of rigid bodies, the connection points between bodies are referred to as joints, where each joint determines the freedom of motion between the 2 connected bodies, also called joint kinematics. To describe the motion, joint models are used: These describe the relative motion between reference frames in different links using joint variables, coordinate transforms, free modes and constrained modes. Joint variables are variables describing the relative motion and pose between 2 links, which can be used to create equations of motion (EoM) specified using joint motion rather than each link's motion relative to the inertial frame. The modes define the directions in which joint motion is allowed and not. The free modes denoted Φ_i for each joint are the directions where motion is allowed.

The free modes can be combined in order to project the system dynamics into the minimum number of joint variables needed to describe the dynamics. Where the projection matrix is decribed by equation 3.1. The constrained modes do not allow motion and will hence not involve any dynamic joint variables, however the corresponding equations can be used to calculate the bearing forces at the joints.

$$\Phi = \text{diag}([\Phi_1, \Phi_2, \dots]) \quad (3.1)$$

Revolute and 6-DOF joints are presented in table 3.1. The 6-DOF joint has no constrained modes and is often called a floating base. Within this thesis, all revolute joints are defined with rotation around the z-axis. See reference [20] for more joint models.

Table 3.1: Joint model definitions

Joint type	Free modes Φ_i	Joint rate variables \dot{q}_i
Revolute	$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$	\dot{q}_i
6-DOF	$I_{6 \times 6}$	$\begin{bmatrix} v_i \\ \omega_i \end{bmatrix}$

3.1.3 Recursive equation of motion

Considering a open chain system, with multiple links/rigid bodies, each individual link can be described by the rigid-body equation of motion shown in equation 2.15. Where each set of equations describes the motion of the specific link's body frame with respect to the inertial NED frame.

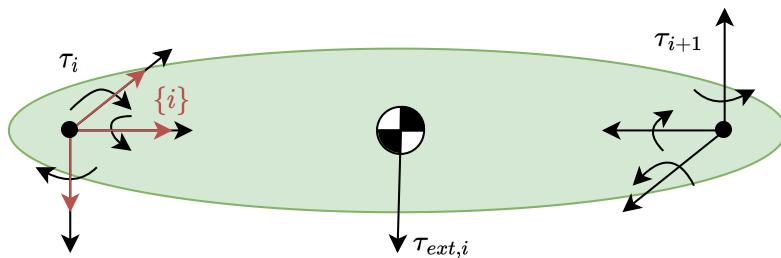


Figure 3.2: Generalized forces acting in a link

The generalized forces acting in each individual link are illustrated in figure 3.2. Which indicates that the sum of generalized forces with respect to the links body frame can be written as a sum of 3 components: The external forces $\tau_{ext,i}$ (including gravity) acting on the link, the forces acting at the joints to the respective successor links $\sum_{j \in s(i)} \tau_j$ and the forces acting at the joint connecting the link to

the predecessor τ_i .

$$\tau_i^i = \tau_i + \sum_{j \in s(i)} \tau_j + \tau_{ext,i} \quad (3.2)$$

Where $\sum_{j \in s(i)} \tau_j$ is the sum over all joints j for all successors of link i ($s(i)$).

Looking at the special case when each link has only one successor, 3.2 can be expressed as equation 3.3. Where τ_{i+1} is the generalized force at the joint connecting the link to the successor, with respect to the successor reference frame $\{i+1\}$. And $K_{i+1}^{\tau i}$ transform this force vector into $\{i\}$ using the coordinate transform. It is important to remember that the forces acting between two links at a common joint will be equal, but of opposite sign.

$$\tau_i^i = \tau_i - K_{i+1}^{\tau i} \tau_{i+1} + \tau_{ext,i} \quad (3.3)$$

Merging the sum of generalized forces 3.3 and the EoM in equation 2.15 result in the iterative equation 3.4. Which is similar to the result found in [21], using the notation from [16] and adding in a extra term for the external forces.

$$M_{RBi} \ddot{\nu}_i + C_{RBi}(\nu_i) \dot{\nu}_i = \tau_i - K_{i+1}^{\tau i} \tau_{i+1} + \tau_{ext,i} \quad (3.4)$$

Where the equation describes the relation between the generalized forces and motion each link experiences, by propagating from the end link towards the base.

Matrix representation

Expanding equation 3.4 into matrix form:

$$M_{RB}^{sys} \begin{bmatrix} \dot{\nu}_1 \\ \dot{\nu}_2 \\ \vdots \\ \dot{\nu}_{sys} \end{bmatrix} + C_{RB}^{sys} \begin{bmatrix} \nu_1 \\ \nu_2 \\ \vdots \\ \nu_{sys} \end{bmatrix} = (I_{6n \times 6n} - W^T(\mathbf{q})) \underbrace{\begin{bmatrix} \tau_1 \\ \tau_2 \\ \vdots \\ \tau_{joints} \end{bmatrix}}_{\tau_{ext}} + \underbrace{\begin{bmatrix} \tau_{ext,1} \\ \tau_{ext,2} \\ \vdots \\ \tau_{ext}^{sys} \end{bmatrix}}_{\tau_{ext}} \quad (3.5a)$$

$$M_{RB}^{sys} = \begin{bmatrix} M_{RB1} & 0_{6 \times 6} & \cdots \\ 0_{6 \times 6} & \ddots & \ddots \\ \vdots & \ddots & M_{RBn} \end{bmatrix} \quad (3.5b)$$

$$C_{RB}^{sys} = \begin{bmatrix} C_{RB1} & 0_{6 \times 6} & \cdots \\ 0_{6 \times 6} & \ddots & \ddots \\ \vdots & \ddots & C_{RBn} \end{bmatrix} \quad (3.5c)$$

$$W^T(\mathbf{q}) = \begin{bmatrix} 0_{6 \times 6} & \tau^1 & 0_{6 \times 6} & \cdots \\ 0_{6 \times 6} & 0_{6 \times 6} & \ddots & \ddots \\ \vdots & \ddots & \ddots & K_n \\ \vdots & \ddots & \ddots & \ddots \end{bmatrix} \quad (3.5d)$$

Where n is the number of links in the multi-link system. Note that the system Coriolis-centripetal matrix is dependent on the generalized velocity, hence the full expression is $C_{RB}^{sys}(\nu_{sys})$.

The matrix representation represents the equation of motion of the entire multi-link system as a function of the force and motion vectors acting in each individual link frame. A representation using the joint variables is often used in robot control and simulation and is developed in section 3.1.5, where the motion vectors are transformed into the joint variables and their derivatives.

Matrices of coordinate transforms

The matrix $W^T(\mathbf{q})$ is of interest as it keeps appearing in the link velocity and acceleration transform. The transposed matrix can be written as equation 3.6 using the property in equation 2.24

$$W(\mathbf{q}) = \begin{bmatrix} 0_{6 \times 6} & 0_{6 \times 6} & 0_{6 \times 6} & \cdots \\ K_1^2 & 0_{6 \times 6} & \ddots & \ddots \\ 0_{6 \times 6} & \ddots & \ddots & \ddots \\ \vdots & \ddots & K_{n-1}^n & \ddots \end{bmatrix} \quad (3.6)$$

Further $W(\mathbf{q})$ is nilpotent as $W(\mathbf{q})^n = 0$, which is a matrix property that imply that the inverse of $(I_{6n \times 6n} - W(\mathbf{q}))^{-1}$ can be written $(I_{6n \times 6n} - W(\mathbf{q}))^{-1} = I + W(\mathbf{q}) + \dots + W(\mathbf{q})^{n-1}$ [21]. Adding these terms together give:

$$\mathcal{L}(\mathbf{q}) = (I_{6n \times 6n} - W(\mathbf{q}))^{-1} = \begin{bmatrix} I_{6 \times 6} & 0_{6 \times 6} & 0_{6 \times 6} & \cdots & \cdots \\ K_1^2 & I_{6 \times 6} & \ddots & \ddots & \ddots \\ K_1^3 & K_2^3 & \ddots & \ddots & \ddots \\ \vdots & \vdots & \ddots & \ddots & \ddots \\ K_1^n & K_2^n & K_3^n & \cdots & I_{6 \times 6} \end{bmatrix} \quad (3.7)$$

Where the transformation elements are produced using the relation in equation 2.25.

In addition, the following relation can be proven by comparing the inverse of $(I_{6n \times 6n} - W^T(\mathbf{q}))$ and the transpose of equation 3.7:

$$\mathcal{L}^T(\mathbf{q}) = (I_{6n \times 6n} - W^T(\mathbf{q}))^{-1} \quad (3.8)$$

3.1.4 Link velocity and acceleration

The velocity and acceleration for each link can be found by iterating from the base toward the end link of the open chain. The result is an iterative expression of the velocity and acceleration expressed in each link's frame, which can be utilized to transform the generalized motion vectors into joint variable functions.

Link velocity

The velocity of a link can be described as the sum of the velocity of the predecessor link and the relative velocity between link i and the predecessor, link $i - 1$.

$$\nu_i = \nu_{i-1} + \nu_{rel} \quad (3.9)$$

Where ν_{rel} is the relative velocity between the two links, which is the joint velocity. Hence the ν_{rel} can be expressed using the joint model and joint velocity. In addition ν_{i-1} is expressed in the predecessor frame, and need to be transformed to $\{\mathbf{i}\}$ using K_{i-1}^i .

$$\nu_i = K_{i-1}^i \nu_{i-1} + \Phi_i \dot{\mathbf{q}}_i \quad (3.10)$$

Expanding the expression into matrix form and moving all frame velocities in the left side give:

$$(I_{6n \times 6n} - W(\mathbf{q})) \mathbf{v}_{sys} = \Phi \dot{\mathbf{q}} \quad (3.11)$$

Finally, using the relation in equation 3.7 the expression for the generalized velocity of all frames in the system can be written:

$$\mathbf{v}_{sys} = \mathcal{L}(\mathbf{q}) \Phi \dot{\mathbf{q}} \quad (3.12)$$

Link acceleration

The link acceleration can be found by taking the derivative of equation 3.9 resulting in equation 3.13. As the coordinate transform is general for all spatial motion vectors, the acceleration in the predecessor frame can be transformed using the same matrix as for the velocity. Together with the derivative of the joint velocity, this becomes equation 3.14. A detailed derivation can be found in reference [20](chapter 2).

$$\dot{\mathbf{v}}_i = \dot{\mathbf{v}}_{i-1} + \dot{\mathbf{v}}_{rel} \quad (3.13)$$

$$\dot{\mathbf{v}}_i = K_{i-1}^i \dot{\mathbf{v}}_{i-1} + \Phi_i \ddot{\mathbf{q}}_i + \dot{\Phi}_i \dot{\mathbf{q}}_i \quad (3.14)$$

Where $\dot{\Phi}_i$ is the derivative of the joint model with respect to time. Which can be expressed as equation 3.15 when Φ_i represent a joint motion axis that is fixed in both link i and $i - 1$, see derivation in [19](chapter 2). This will be the case for both rational and 6DOF joints.

$$\dot{\Phi}_i = S^s(\mathbf{v}_i) \Phi_i \quad (3.15)$$

Rearranging the equation and expanding to matrix form:

$$(I - W(\mathbf{q})) \dot{\mathbf{v}}_{sys} = \Phi \ddot{\mathbf{q}} + \begin{bmatrix} S^s(\mathbf{v}_1) \Phi_1 & 0_{6 \times 6} & \cdots \\ 0_{6 \times 6} & \ddots & \ddots \\ \vdots & \ddots & S^s(\mathbf{v}_n) \Phi_n \end{bmatrix} \dot{\mathbf{q}} \quad (3.16)$$

Notice that the spatial cross product will be reduced when inserting for \mathbf{v}_i using equation 3.10. Due to the second term of $S^s(K_{i-1}^i \mathbf{v}_{i-1} + \Phi_i \dot{\mathbf{q}}_i) \Phi_i \dot{\mathbf{q}}_i$ becomes zero by definition of the spatial cross product as the vectors are parallel. Using this property and changing the order of the spatial motion cross product:

$$(I - W(\mathbf{q}))\dot{\mathbf{v}}_{sys} = \Phi\ddot{\mathbf{q}} - S^I(\Phi\dot{\mathbf{q}}) \left(W(\mathbf{q}) \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} + \underbrace{\begin{bmatrix} K_0^1 v_0 \\ 0_{6 \times 1} \\ \vdots \\ v_{base} \end{bmatrix}}_{v_{base}} \right) \quad (3.17a)$$

$$S^I(\Phi\dot{\mathbf{q}}) = \begin{bmatrix} S^s(\Phi_1\dot{\mathbf{q}}_1) & 0_{6 \times 6} & \cdots & \\ 0_{6 \times 6} & \ddots & \ddots & \\ \vdots & \ddots & S^s(\Phi_n\dot{\mathbf{q}}_n) & \end{bmatrix} \quad (3.17b)$$

Finally, the expression for the acceleration can be written as equation 3.18, when equation 3.12 for the velocity is inserted.

$$\ddot{\mathbf{v}}_{sys} = \mathcal{L}(\mathbf{q})(\Phi\ddot{\mathbf{q}} - S^I(\Phi\dot{\mathbf{q}})(W(\mathbf{q})\mathcal{L}(\mathbf{q})\Phi\dot{\mathbf{q}} + v_{base})) \quad (3.18)$$

Which is only dependent on the states \mathbf{q} and the velocity of the base. Within this thesis, the base will be the NED frame, and its acceleration and velocity will be zero.

3.1.5 Equation of motion in closed form

The equation of motion can be expressed on a closed matrix form by utilizing the matrix form developed in section 3.1.3 together with the velocity and acceleration presented in section 3.1.4. More specific equation 3.5, 3.12 and 3.13 are combined to create a model that only dependent on the joint states, which can be written on the from:

$$\mathcal{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathcal{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} = \mathcal{T} \quad (3.19)$$

Notice the difference between the generalized force vector τ and the joint forces \mathcal{T} , which contain an input for every joint state the system has.

The matrix form in equation 3.5 express the generalized forces experienced in each joint for all six degrees of freedom. This is useful for structural analysis, but for control, the focus is on the joint variables as this gives the minimum representation of the dynamic system. The relevant equations can be extracted using the free modes of the kinematic joints.

$$\mathcal{T} = \Phi^T \tau_{joints} = \Phi^T \mathcal{L}^T(\mathbf{q})(M_{RB}^{sys} \dot{\mathbf{v}}_{sys} + C_{RB}^{sys}(\mathbf{v}_{sys})\mathbf{v}_{sys} - \tau_{ext}^{sys}) \quad (3.20)$$

Assuming the base frame is stationary ($v_{base} = 0$ and $\dot{v}_{base} = 0$) and inserting for the acceleration and velocity vectors result in the final result:

$$\mathcal{T} = \mathcal{M}(q)\ddot{q} + \mathcal{C}(q, \dot{q})\dot{q} - \Phi^T \mathcal{L}^T(q) \tau_{ext}^{sys} \quad (3.21a)$$

$$\mathcal{M}(q) = \Phi^T \mathcal{L}^T(q) M_{RB}^{sys} \mathcal{L}(q) \Phi \quad (3.21b)$$

$$\mathcal{C}(q, \dot{q}) = \Phi^T \mathcal{L}^T(q) (-M_{RB}^{sys} \mathcal{L}(q) S^I(\Phi \dot{q}) W(q) + C_{RB}^{sys}) \mathcal{L}(q) \Phi \quad (3.21c)$$

Where C_{RB}^{sys} is short for the Coriolis-centripital matrix expressed using joint motion $C_{RB}^{sys}(\mathcal{L}(q)\Phi \dot{q})$.

Notice how τ_{ext}^{sys} allows the external forces to be expressed within each individual link, which can be exploited when adding the forces due to gravity, aerodynamic forces, or others. The gravity forces are state-dependent and are often expressed separately. However, here they are included in the external forces.

The final expression in equation 3.21 is similar to the end result found in reference [21]. However, they differ due to the added external force term in this thesis. In addition, the forces due to gravity are added by accelerating the base (link 0) in reference [21] as a alternative to introducing them using τ_{ext}^{sys} .

Notice the difference between M, C, τ, ν, η and $\mathcal{M}, \mathcal{C}, \mathcal{T}, q$, where the first set is related to a single body with 6 or less degrees of freedom. While the second set describes a multi-body system with one or more degrees of freedom using joint variables.

3.2 Surface vessel model

In this section, a surface vessel maneuvering model is presented. The model is adopted from reference [16] which is Fossen's robot-inspired model for marine crafts, which has become the de facto standard for marine craft modeling.

3.2.1 Maneuvering model

To describe the dynamics of the craft, a nonlinear ship maneuvering model derived in reference [16] is adopted, and the coming section is based on the material found there. It is common to use a 3 or 4DOF representation, however a 6DOF model is presented here as it will be merged with the open chain model later. Notice that the keel and rudder forces enter the model through the generalized force τ and will be presented in section 4.1.5.

The maneuvering model is represented by the equation:

$$\underbrace{M_{RB} \dot{\nu} + C_{RB}(\nu) \nu}_{\text{Rigid-body forces}} + \underbrace{M_A \dot{\nu}_{rc} + C_A(\nu_{rc}) \nu_{rc} + D(\nu_{rc}) \nu_{rc}}_{\text{Hydrodynamic forces}} + \underbrace{g(\eta)}_{\text{Hydrostatic forces}} = \tau \quad (3.22)$$

Where one can recognize the system inertia matrix and the Coriolis-centripetal matrix from equation 2.15. While the hydrodynamic and hydrostatic forces are caused by interaction with the water/fluid surrounding the vessel. To incorporate the hydrodynamic effects of the ocean current, the hydrodynamic forces are functions of the velocity of the vessel relative to the ocean current, which is the relative velocity the vessel hull experience. The relative velocity becomes:

$$\nu_{rc} = \nu - \nu_c \quad (3.23)$$

Where ν_c is the velocity of the current expressed in $\{b\}$.

The new matrices describe:

M_A The added mass matrix, which represent a virtual mass added to the system as the vessel body has to move some volume of the surrounding fluid as it accelerate and decelerate [16].

$C_A(\nu_{rc})$ The Coriolis and centripetal matrix caused by the the added mass.

$D(\nu_{rc})$ Damping matrix.

$g(\eta)$ Generalized force vector caused by buoyancy and gravity acting on the vessel.

It can be shown that the model can be simplified to equation 3.24 if the ocean current is assumed to be constant and irrotational in $\{n\}$ and the rigid-body Coriolis-centripetal matrix is independent of the linear velocity [16]. The current velocity vector can be written as $\nu_c = [u_c \ v_c \ w_c \ 0 \ 0 \ 0]^T$. While the rigid-body Coriolis-centripetal matrix found in equation 2.15c is independent of the linear velocity. M is the system inertia matrix.

$$\underbrace{(M_{RB} + M_A)}_M \dot{\nu}_{rc} + \underbrace{(C_{RB}(\nu_{rc}) + C_A(\nu_{rc}))}_{C(\nu_{rc})} \nu_{rc} + D(\nu_{rc}) \nu_{rc} + g(\eta) = \tau \quad (3.24)$$

Notice that even with constant and irrotational current, the current velocity derivative is not equal to zero, as the velocity states are expressed in the rotating reference frame $\{b\}$:

$$\dot{\nu}_c = \begin{bmatrix} -S(\omega_{nb}^b) & 0_{3x3} \\ 0_{3x3} & 0_{3x3} \end{bmatrix} \nu_c \quad (3.25)$$

Combining equation 3.24 and 3.25 the state-space model can be expressed using absolute velocity:

$$\dot{\nu} = -\dot{\nu}_c + M^{-1}(\tau - N(\nu_{rc})\nu_{rc} - g(\eta)) \quad (3.26a)$$

$$N(\nu_{rc}) = C_{RB}(\nu_{rc}) + C_A(\nu_{rc}) + D(\nu_{rc}) \quad (3.26b)$$

Notice that the kinematics of the model expressed in equation 2.8 is dependent on the absolute velocity. Hence if relative velocity is used as the system state it needs to be accounted for in the kinematic equation by applying the relation in equation 3.23.

3.2.2 Hydrodynamic forces

The coefficients of the added mass matrix and damping matrix can be estimated by using computer programs. The matrices are frequency dependent, however in maneuvering theory, they are assumed constant at zero-frequency for the surge, sway and yaw (mass-damper) dynamics. Where it is assumed that the zero-frequency matrices approximate the dominating dynamics. For the second-order mass-damper-spring systems; heave, roll and pitch, the coefficients are selected at their respective natural frequency [16].

Added mass and Coriolis-centripetal matrices

The added mass matrix is defined:

$$M_A = - \begin{bmatrix} X_{\dot{u}} & X_{\dot{v}} & X_{\dot{w}} & X_{\dot{p}} & X_{\dot{q}} & X_{\dot{r}} \\ Y_{\dot{u}} & Y_{\dot{v}} & Y_{\dot{w}} & Y_{\dot{p}} & Y_{\dot{q}} & Y_{\dot{r}} \\ Z_{\dot{u}} & Z_{\dot{v}} & Z_{\dot{w}} & Z_{\dot{p}} & Z_{\dot{q}} & Z_{\dot{r}} \\ K_{\dot{u}} & K_{\dot{v}} & K_{\dot{w}} & K_{\dot{p}} & K_{\dot{q}} & K_{\dot{r}} \\ M_{\dot{u}} & M_{\dot{v}} & M_{\dot{w}} & M_{\dot{p}} & M_{\dot{q}} & M_{\dot{r}} \\ N_{\dot{u}} & N_{\dot{v}} & N_{\dot{w}} & N_{\dot{p}} & N_{\dot{q}} & N_{\dot{r}} \end{bmatrix} \quad (3.27)$$

The hydrodynamic Coriolis and centripetal matrix $C_A(\nu_{rc})$ can be found similarly to C_{RB} , by using the added mass matrix. $C_A(\nu_{rc})$ can be parameterized as shown in equation 3.28, by sectioning the added mass matrix as in equation 3.29.

$$C_A(\nu) = \begin{bmatrix} 0_{3 \times 3} & -S(A_{11}\nu_1 + A_{12}\nu_2) \\ -S(A_{11}\nu_1 + A_{12}\nu_2) & -S(A_{21}\nu_1 + A_{22}\nu_2) \end{bmatrix} \quad (3.28)$$

$$M_A := \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \quad (3.29)$$

Where A_{xx} refers to the 3×3 sub-matrices of M_A , while ν_1 and ν_2 refers to the linear and angular velocity respectively.

Damping matrices

The damping terms contribute to both linear and quadratic damping, hence these can be expressed as two terms even though they can be difficult to separate in practice. Where the linear damping is caused by linear potential- and viscous-damping [16]. The nonlinear effect will here be based on the nonlinear surge damping and cross-flow drag, which are assumed to be the dominating nonlinear effects.

$$D(\nu_{rc})\nu_{rc} = D\nu_{rc} + d(\nu_{rc}) \quad (3.30)$$

As mentioned, the damping matrices can be found using hydrodynamic computer programs; an alternative is to apply hydrodynamic theory directly, which is the approach used in this thesis. For simplicity, the linear damping matrix is assumed to be diagonal:

$$D = -\text{diag}([X_u, Y_v, Z_w, K_p, M_q, N_r]) \quad (3.31)$$

At zero frequency the potential damping is zero; hence viscous damping will be dominant, which is also true at low frequencies [16]. The linear surge, sway and yaw damping can hence be described by the viscous damping at zero frequency. In reference [16] it is suggested to select the viscous damping by using time constants T for the 3 mass-damper systems surge, sway and yaw. As viscous damping is dominant, this element can be used to approximate the elements in the linear damping matrix:

$$-X_u = \frac{m - X_{\dot{u}}}{T_1} \quad (3.32)$$

$$-Y_v = \frac{m - Y_{\dot{v}}}{T_2} \quad (3.33)$$

$$-N_r = \frac{I_{zz}^{CO} - N_{\dot{r}}}{T_3} \quad (3.34)$$

Where the numerators are diagonal elements of the total system inertia matrix $M = M_{RB} + M_A$.

In the same reference, it is suggested to approximate the linear damping in heave, roll and pitch by specifying 3 damping ratios ζ .

$$-Z_w = 2\zeta_3 \omega_3 M_{33} \quad (3.35)$$

$$-K_p = 2\zeta_4 \omega_4 M_{44} \quad (3.36)$$

$$-M_q = 2\zeta_5 \omega_5 M_{55} \quad (3.37)$$

Where ω_x is the natural frequency and M_{xx} are diagonal elements of the total system inertia matrix.

The nonlinear damping will dominate at high velocity due to the quadratic and cross terms, while linear damping will dominate at low velocity. The heave, roll and pitch velocities are assumed to be small; hence their nonlinear contributions are neglected. The nonlinear damping can hence be written :

$$d(v_{rc}) = -[X_{nsd} \quad Y_{cf_d} \quad 0 \quad 0 \quad 0 \quad N_{cf_d}]^T \quad (3.38)$$

The first term is the nonlinear surge damping X_{nsd} which can be described by equations 3.39 [30]. Where k_f represent the form factor viscous correction, which is typically 0.1 for a ship in transit. S_h is the wetted surface of the hull and C_R is the residual friction. R_n is the Reynolds number, which is calculated using the length of the hull L_h and the kinematic viscosity $\nu_{kv} = 1 \times 10^{-6} \text{ m/s}^2$ (at 20°C). \mathcal{E} is a small constant introduced to obtain a finite solution for all u_{rc} .

$$X_{nsd} = -\frac{1}{2}\rho S_h(1 + k_f)C_f(u_{rc})|u_{rc}|u_{rc} \quad (3.39a)$$

$$C_f(u_{rc}) = \frac{0.075}{(\log_{10}(R_n) - 2)^2 + \mathcal{E}} + C_R \quad (3.39b)$$

$$R_n = \frac{L_h}{\nu_{kv}} |u_{rc}| \geq 0 \quad (3.39c)$$

Y_{cf_d} and N_{cf_d} are the nonlinear cross-flow drag terms which can be calculated using equation 3.40 and 3.41 respectively [31]. Where x represent the x-position along the hull with respect to the midpoint. $T(x)$ is the hull draft as function of the x-position. $C_d^{2D}(x)$ is the 2D drag coefficient calculated using the Hoerner's curve on page 155 in reference [16]. Which is based on stripe theory, where the integrals are evaluated over the all yz- cross sections along the hull.

$$Y_{cf_d} = -\frac{1}{2}\rho \int_{-\frac{L_h}{2}}^{\frac{L_h}{2}} T(x)C_d^{2D}(x)|v_{rc} + xr|(v_{rc} + xr)dx \quad (3.40)$$

$$N_{cf_d} = -\frac{1}{2}\rho \int_{-\frac{L_h}{2}}^{\frac{L_h}{2}} T(x)C_d^{2D}(x)x|v_{rc} + xr|(v_{rc} + xr)dx \quad (3.41)$$

3.2.3 Hydrostatic forces

For a floating, metacentric stable vessel at equilibrium, the buoyancy and the gravitational forces and moments cancel. Where metacentric stable means that for a small perturbation from the equilibrium in heave, roll or pitch, the resulting forces and moments will try to resist the perturbation. Due to spring stiffness pulling the system back to equilibrium, these are often referred to as restoring forces.

For a floating vessel equation 3.42 holds. Where ∇ is the volume of water displaced by the vessel.

$$mg = \rho g \nabla \quad (3.42)$$

A linear expression for the restoring forces matrix is shown in equation 3.43, which visualize the spring stiffness in the center of flotation CF (geometric center of A_{wp}). The derivation of the matrix can be found in reference [16] chapter 4, where the following assumptions are used: The perturbations are small and the waterplane is A_{wp} is constant.

$$G^{CF} = \text{diag}([0, 0, \rho g A_{wp}, \rho g \nabla GM_T, \rho g \nabla GM_L, 0]) \quad (3.43)$$

Where GM_T is the transverse metacentric height and GM_L is the longitudinal metacentric height. These describe the distance from CG to the respective metacenter. Where the perturbation angle times the metacentric height becomes the moment-arm of the restoring moment. For a metacentric stable vessel, it is required that both metacentric heights are positive and above a desirable minimum limit.

The hydrostatic force can be described by transforming G^{CF} to the CO :

$$g(\eta) = G\eta = H^T(r_{bf}^b)G^{CF}H(r_{bf}^b)\eta \quad (3.44)$$

Where r_{bf}^b is the vector from CO to CF in $\{\mathbf{b}\}$.

Metacentric height calculation

The metacentric height can be expressed:

$$GM_T = \frac{I_T}{\nabla} + r_{bCG,z}^b - r_{bCB,z}^b \quad (3.45)$$

$$GM_L = \frac{I_L}{\nabla} + r_{bCG,z}^b - r_{bCB,z}^b \quad (3.46)$$

Where it assumed that CG and CB lie on the same vertical line. Hence only the z-component of the vectors from CO to CG and CB contribute. With the center of buoyancy CB located at the volume center of the submerged volume of the saildrone. I_T and I_L are the second moment of areas with respect to A_{wp} .

Chapter **4**

Modeling

In this chapter, multiple models are developed representing the dynamics of the saildrone. First, a simulation model is developed combining the open chain model and the surface vessel model presented in chapter 3. Then a simplified model for control is derived by system simplification. In the end, the open loop responses of the models are compared in order to verify that the simplified model is representative.

Thought the chapter both 6-, 4- and 3DOFs are used. The simulation model is 6DOF due to the convenience of spatial algebra and the robot chain formulation. However, in this study effects of waves are omitted; hence the heave and pitch motion will be neglectable. Therefore the simplified models are represented as 4- and 3DOFs.

The states of the saildrone are given by the state vector $\bar{q}^* = [\eta \quad \varepsilon \quad \delta_t \quad \delta_r]^T$ and the velocity state vector $\dot{q}^* = [\nu \quad \dot{\varepsilon} \quad \dot{\delta}_t \quad \dot{\delta}_r]^T$ which are further described in section 4.1.1. The rudder dynamics are modeled separately hence when it is included in the state vector the superscript * is used. Figure 4.1 show how the most of the states are defined. Where positive rotation follow the right hand rule.

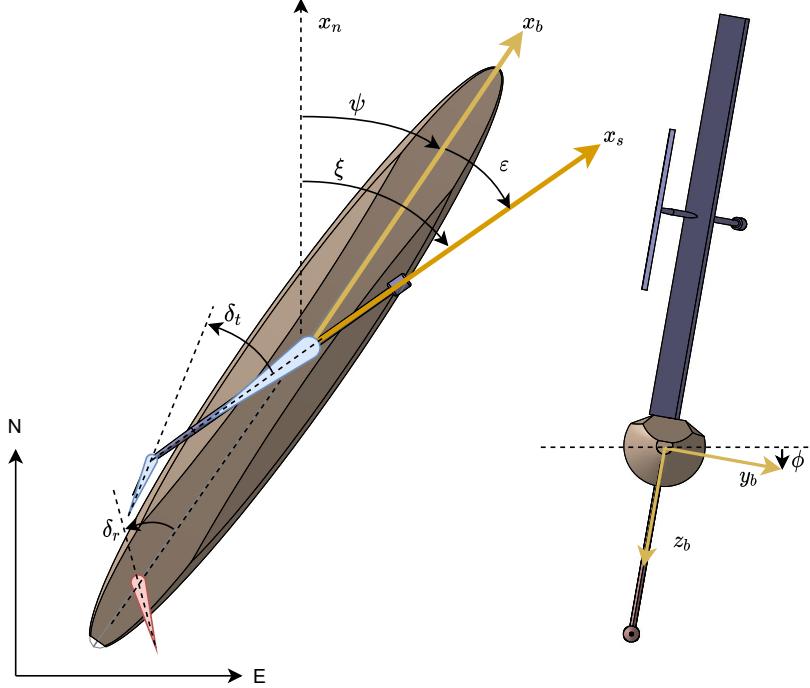


Figure 4.1: Saildrone states

4.1 Simulation model

In this section, the robot-inspired model for marine craft from section 3.2 with the open chain model in section 3.1 are merged to form a simulation model of the multi-link system the saildrone represent. The model is created by first using an open chain formulation for the vessel, sail and tail. Then applying the hydrodynamic and hydrostatic forces from the vessel hull, using the surface vessel model. The generalized force induced by the keel is not accounted for in the vessel model and is hence introduced as an external force acting on the vessel link. Finally, the rudder is added to the model, here a simple first-order model is used under the assumption that there exists a lower-level controller for the rudder angle. The rudder inertia and mass are assumed neglectable, and the resulting foil-induced forces and moments are transformed into the body frame using a coordinate transform.

4.1.1 Merging the robot and vessel model

The saildrone is modeled based on the open chain model, where the NED frame $\{n\}$ is the inertial frame denoted $\{0\}$, the vessel body frame $\{b\}$ is the first frame, the sail $\{s\}$ the second and the tail $\{t\}$ is the third frame following the numbering from section 3.1. The relation between $\{n\}$ and $\{b\}$ is a 6DOF floating base joint,

while the vessel and sail are connected through a 1DOF revolute joint. In the same way, the sail and tail are also connected through a revolute joint. The frames are defined at the joints. Hence they are placed as shown in figure 4.2, which also shows the keel and rudder frames.

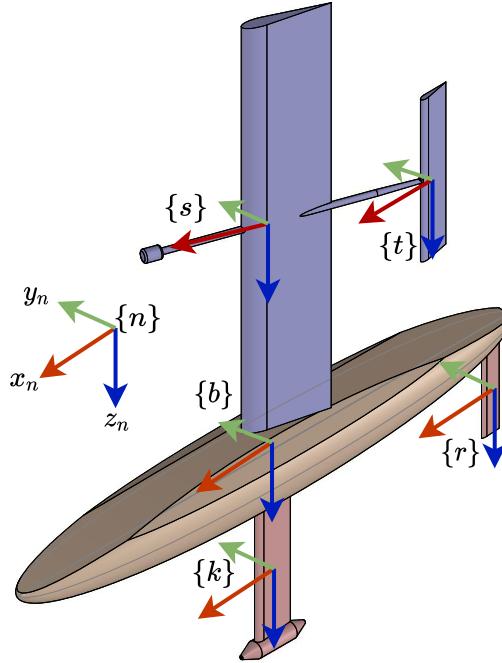


Figure 4.2: Saildrone frames

Using the frames defined in figure 4.2, the relation between the vessel and $\{n\}$ is identical to the surface vessel model presented in section 3.2. Meaning the hydrodynamic and hydrostatic forces and moments can be introduced directly in $\{b\}$. From the robot model perspective, these are external forces and moments acting on a link and can hence be introduced directly. Notice that the vessel model does not include the keel forces and moments; therefore, these need to be added separately. Further, the generalized forces introduced by the rudder need to be introduced.

For the 3-link system consisting of vessel, SAW and tail, the equation of motion can be written using equation 4.1. Where \mathcal{T} are the joint forces for the free modes and $\tau_{ext,i}^i$ are the generalized external forces in each link i . Notice that \bar{q} is introduced as it is desirable to express the body's position and attitude in $\{n\}$. However, the joint variables for the sail and tail angle are left unchanged.

$$\mathcal{M}(\bar{\mathbf{q}})\ddot{\mathbf{q}} + \mathcal{C}(\bar{\mathbf{q}}, \dot{\mathbf{q}})\dot{\mathbf{q}} = \mathcal{T} + \Phi^T \mathcal{L}^T(\bar{\mathbf{q}}) \begin{bmatrix} \tau_{ext,b}^b \\ \tau_{ext,s}^s \\ \tau_{ext,t}^t \end{bmatrix} \quad (4.1a)$$

$$\bar{\mathbf{q}} = \begin{bmatrix} \bar{q}_1 \\ \bar{q}_2 \\ \bar{q}_3 \end{bmatrix} = \begin{bmatrix} \eta \\ \varepsilon \\ \delta_t \end{bmatrix} \quad (4.1b)$$

$$\dot{\mathbf{q}} = \underbrace{\begin{bmatrix} J_\Theta(\eta) & 0_{6 \times 1} & 0_{6 \times 1} \\ 0_{1 \times 6} & 1 & 0 \\ 0_{1 \times 6} & 0 & 1 \end{bmatrix}}_{\dot{\mathbf{q}}} \begin{bmatrix} \nu \\ \dot{\varepsilon} \\ \dot{\delta}_t \end{bmatrix} \quad (4.1c)$$

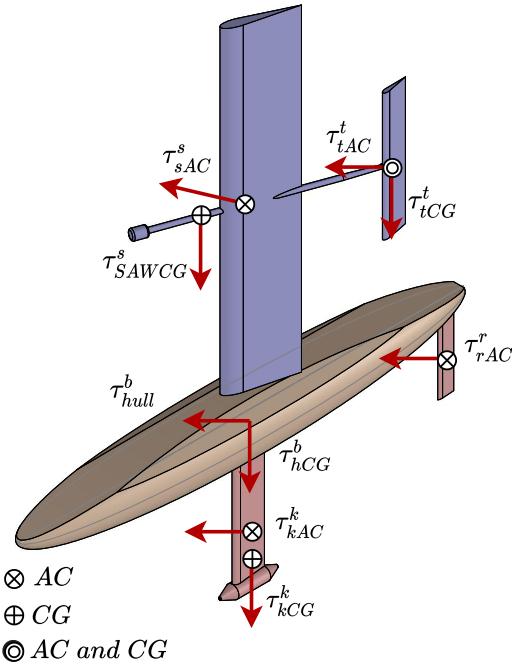


Figure 4.3: Generalized forces acting on the saildrone

In the proposed EoM in equation 4.1 the generalized forces due to the rudder and keel are not written explicit as these are introduced through the generalized force acting on the vessel body. The rudder deflection is modeled as a first-order system, assuming the rudder actuator has an angle controller.

The generalized external forces acting on the saildrone are illustrated in figure 4.3. These are generalized forces acting on different links due to gravity and foil-

and hull-induced forces. τ_{iAC}^i represent the foil-induced forces acting in link i 's AC with respect to the axes of frame i . Similarly, τ_{iCG}^i represents the forces due to gravity acting on the link's center of gravity. τ_{hull}^b are the forces acting on the hull given in the body frame, including both hydrodynamic and hydrostatic buoyancy forces. The mass of the rudder is assumed to be neglectable and hence the rudder has no external force due to gravity.

It is important to notice that generalized forces are of 6 dimensions, while the illustration shows these as 3-dimensional vectors to reduce the number of vectors.

4.1.2 Vessel model contribution

Having established that the hydrodynamic and hydrostatic forces in the vessel model can be introduced directly in equation 4.1. It needs to be determined where these forces enter in the open chain formulation of the EoM. Note that the entire vessel is modeled as a single link, hence the keel and rudder forces will appear as part of the external forces acting on the vessel body.

The full expression for the generalized external forces acting on the vessel body can be found by first writing the isolated vessel model on a general form:

$$M(\dot{v} - \dot{v}_c) + N(v_{rc})v_{rc} + g(\eta) = \tau \quad (4.2)$$

Then rearranging the equation and removing $M\dot{v}$ as it will be a part of $\mathcal{M}(\bar{q})\ddot{q}$, result in the external forces acting on the vessel body link. Hence the external force $\tau_{ext,b}^b$ from equation 4.1a can be written as equation 4.3.

$$\tau_{ext,b}^b = \underbrace{M\dot{v}_c - N(v_{rc})v_{rc} - g(\eta)}_{\tau_{hull}^b + \tau_{hCG}^b + \tau_{kCG}^b} + \tau_{kAC}^b + \tau_{rAC}^b \quad (4.3)$$

Where all the generalized forces acting on the hull, keel and rudder are transformed to CO. The transforms for the keel and rudder forces are shown in equation 4.4 - 4.6. These consist of two transforms each. The first is from the point of action in their respective frame, to the frame's origin. Second, the forces are transformed from the frame to the body frame using coordinate transforms for generalized forces.

$$\tau_{kAC}^b = K_k^{b\tau k} K_{kCG}^{\tau k} \tau_{kCG}^k \quad (4.4)$$

$$\tau_{kAC}^b = K_k^{b\tau k} K_{kAC}^{\tau k} \tau_{kAC}^k \quad (4.5)$$

$$\tau_{kAC}^b = K_r^{b\tau r} K_{rAC}^{\tau r} \tau_{rAC}^r \quad (4.6)$$

Notice that $N(\nu_{rc})$ contain the Coriolis-centripetal matrix for the vessel, which is excluded from $\mathcal{C}(\dot{\mathbf{q}}, \ddot{\mathbf{q}})$ as the term is included in the external forces instead. This is valid due to the following observation, by expanding $\Phi^T \mathcal{L}^T(\mathbf{q})$ it can be verified that it only transform generalized forces from the successor links. Hence forces applied on the vessel will not influence the other links directly. Similarly, by expanding and $\mathcal{C}(\dot{\mathbf{q}}, \ddot{\mathbf{q}})\dot{\mathbf{q}}$ with respect to C_{RB}^{sys} shown in equation 4.7, it can be proved that the Coriolis-centripetal matrix for the first link do not influence the other links directly. This result means that the Coriolis-centripetal matrix of the vessel (first link) can be introduced either in $\mathcal{C}(\dot{\mathbf{q}}, \ddot{\mathbf{q}})$ or as an external force acting on the first link. This justify the use of $N(\nu_{rc})$ in equation 4.3.

$$\Phi^T \mathcal{L}^T(\mathbf{q}) C_{RB}^{sys}(\nu_{sys}) \mathcal{L}(\mathbf{q}) \Phi \quad (4.7)$$

4.1.3 Sail system contribution

The external forces acting on the SAW link are shown in figure 4.3. These are the forces due to gravity and the forces induced by the sail, acting in SAW CG and sail AC, respectively. SAW is defined using the sail frame, hence the forces are simply described in $\{\mathbf{s}\}$. However it is important to use the inertia and mass of the full SAW link in $\mathcal{M}(\ddot{\mathbf{q}})\ddot{\mathbf{q}}$ and $\mathcal{C}(\ddot{\mathbf{q}}, \ddot{\mathbf{q}})\dot{\mathbf{q}}$. Transforming the generalized forces to the sail frame origin and summarizing result in:

$$\tau_{ext,s}^s = K_{sAC}^s \tau_{sAC}^s + K_{SAWCG}^s \tau_{SAWCG}^s \quad (4.8)$$

Where these transforms, that operate between 2 point in the same frame, will be static.

Similar to SAW, the external forces acting on the tail link can be described by the following equation:

$$\tau_{ext,t}^t = K_{tAC}^t \tau_{tAC}^t + K_{tCG}^t \tau_{tCG}^t \quad (4.9)$$

4.1.4 Joint forces

\mathcal{T} is the joint forces input vector used to control the relative motion between joints in an open chain. For the saildrone model, most of the elements of \mathcal{T} are zero.

The 6DOF joint, which the floating base represents, does not have any actuator besides what is already added to the external forces described in section 4.1.2. Hence the first 6 elements of \mathcal{T} are zero.

For the second joint connecting the sail system and the vessel, there are no actuator. The input can be used to add friction models to the revolute joint. However,

in this thesis the joint is assumed to be frictionless. Meaning the seventh element of \mathcal{T} is zero as well.

The last element of \mathcal{T} is the moment input from the tail deflection actuator, which will be controlled to obtain the desired tail angle.

$$\mathcal{T} = \begin{bmatrix} 0_{1 \times 6} & 0 & M_z^t \end{bmatrix}^T \quad (4.10)$$

4.1.5 Foil induced forces and gravity

The generalized force due to gravity acting on a link's center of gravity is formulated τ_{iCG}^i . Gravity will not cause moment in CG hence the vector can be represented by:

$$\tau_{iCG}^i = \begin{bmatrix} R_n^i \mathcal{G}^n m_i \\ 0_{3 \times 1} \end{bmatrix} \quad (4.11)$$

$$\mathcal{G}^n = \begin{bmatrix} 0 & 0 & g \end{bmatrix}^T \quad (4.12)$$

Where m_i is the mass of the link, \mathcal{G}^n is the gravity vector in $\{\mathbf{n}\}$ and R_n^i is the rotation matrix transforming the vector into the link's reference frame. Notice that for the vessel model, the forces due to gravity are included in the calculation of the metacentric stability and are hence included in the restoring forces.

The foil-induced forces and moments acting in the AC are expressed by extracting the components of equation 2.39, which give:

$$\tau_{iAC}^i = \begin{bmatrix} F_x^i & F_y^i & 0 & 0 & 0 & M_z^i \end{bmatrix}^T \quad (4.13)$$

Where the superscript is change from foil to i .

4.1.6 Velocity triangle and angle of attack

Figure 4.4 show the two of the saildrone's velocity triangles, which are the vector relations between the relative velocity, the flow velocity and the velocity with respect to $\{\mathbf{n}\}$.

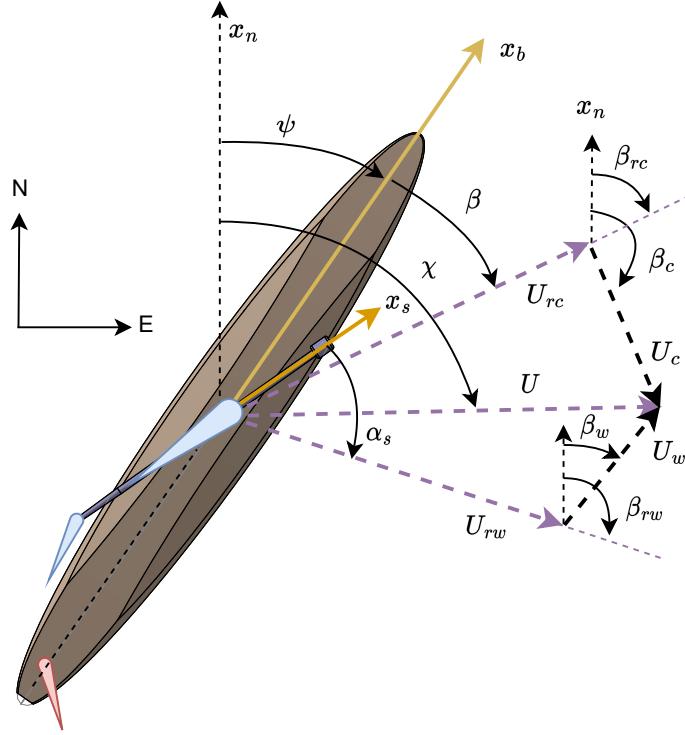


Figure 4.4: Velocity triangles

χ represents the course angle relative to NED. β_c and β_w describe the angle of the current and wind velocity with respect to the NED frame. While β_{rc} and β_{rw} describe the angle of the relative velocity vectors. β is the sideslip angle, which is defined similar to the AOA, but with respect to the CO. α_s is the sail AOA. The magnitude of the velocity vectors is denoted by U , where U_c is the current velocity, U_w is the wind velocity and U is the velocity of the reference frame relative to $\{n\}$. The relative velocities vectors for the body and sail are given by:

$$v_{rc}^b = v^b - v_c^b \quad (4.14)$$

$$v_{rw}^s = v^s - v_w^s \quad (4.15)$$

Angle of attack

Using the relative (linear) velocity at each link's AC, the angle of attack for each link can simply be calculated by:

$$\alpha_i = \text{atan2}\left(\frac{v_r^{iAC}}{u_r^{iAC}}\right) \quad (4.16)$$

Where v_r^{iAC} and u_r^{iAC} are the relative velocity components in x and y (within frame i 's AC), with respect to either wind or current. Where the components of the flow

velocity are found by transforming the flow's velocity vector in NED into the respective links.

The link velocities expressed relative to NED instead of as joint rates, can be found by using equation 3.12 or 3.10, which further can be transformed to the link's AC using the coordinate transform for motion vectors. Equation 3.10 also apply for the keel and rudder where \dot{q}_i is zero for the keel and replaced by $\dot{\delta}_r$ for the rudder.

4.1.7 Rudder model

The rudder deflection angle is modeled as a first-order system with time constant T_r , where it is assumed that the rudder actuator has a suitable angle controller. The system is represented with a saturation limits for the deflection angle, as shown in figure 4.5.

In addition, for the calculation of the rudder forces the rudder rate is used, hence the desired rudder angle δ_{rd} is low-pass filtered to avoid large rudder rate spikes. Her a first order filter with a time constant ten times faster than the rudder angle dynamics is used. The full system can be seen in figure 4.5. A gain of -1 is introduced such that a positive rudder command results in a positive turning rate.

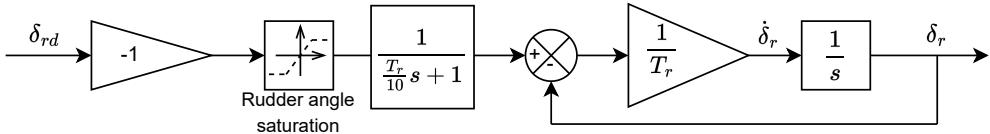


Figure 4.5: Rudder dynamics block diagram

4.1.8 Saildrone dynamics overview

The previous sections can be summarized into the saildrone dynamics shown in figure 4.6. The hydrodynamic, hydrostatic and gravity generalized forces are a part of the open chain dynamics. While the rudder dynamics and the foil-induced forces are separated out.

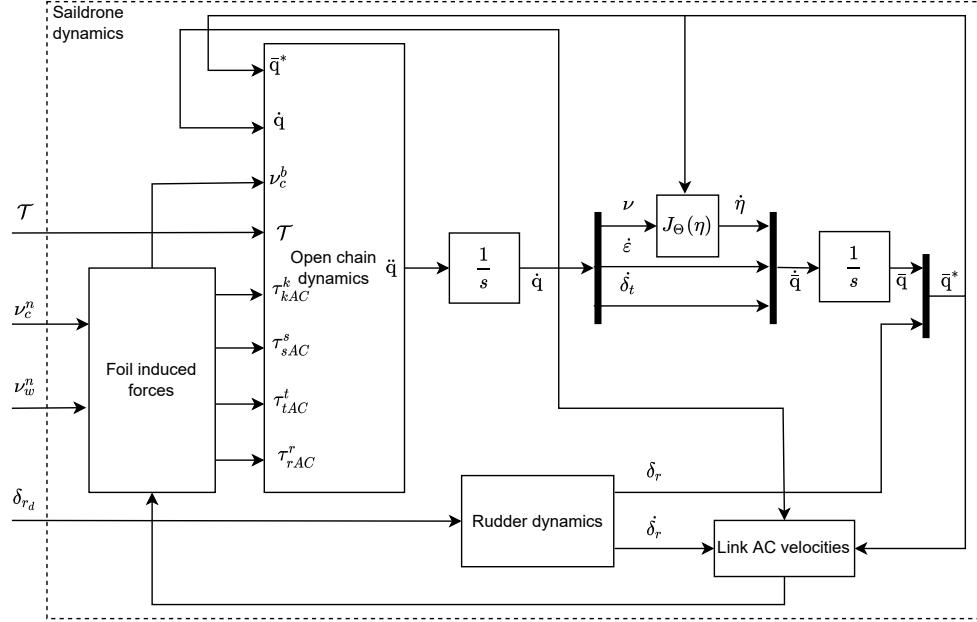


Figure 4.6: Saildrone dynamics block diagram

4.2 Simplified model

In this section, a simplified model of the system dynamics is derived, where the sail system and the vessel are decoupled such that the sail system only provides propulsion to the vessel. The simplified 4 DOF vessel model is based on the vessel model presented in section 3.2, together with the simplified keel and rudder contribution. Where the following state vector is used $\nu_{rc} = [u_{rc} \ v_{rc} \ p \ r]^T$. While the sail system is modeled with one degree of freedom where the state is the heading of the sail system. The rudder and tail angles are given by two first-order models. The simplified model is utilized in the controller- and navigation- design.

4.2.1 Assumptions

In order to justify the decoupling and the use of the simpler vessel model presented in equation 3.24 instead of the highly nonlinear model presented in equation 4.1, a set of design assumptions are introduced:

Foils

- In general all foils are considered to be symmetric, causing no moment around the AC, while the induced forces will be symmetric for positive and negative angle of attack.

Sail system

1. The center of mass of the full sail system, including sail, tail and counterweight, is located at the pivot-point of the sail system. This is reasonable with respect to the design of the saildrone as it implies that CG of the sail system will not cause any moment affecting the sail system heading, independent of the attitude of the saildrone. This is desirable as the moment due to gravity is zero, and no controller compensation is needed.
2. CG of the sail system remains fixed independent of the tail angle. The combined weight of the sail and counterweight is assumed to be greater than the weight of the tail by such a margin that the movement of the CG due to the tail angle is neglectable.
3. The sail system pivot point is assumed to be frictionless, hence no moment is transferred between the sail system and the vessel through the rotational axis of the sail with respect to the vessel. This implies that only forces are transferred and that the inertia, Coriolis and centripetal terms with respect to the sail system's rotational/joint axis do not affect the inertia of the vessel. However, the transferred inertia, Coriolis and centripetal terms with respect to rotation around the x and y axis will be transferred to the vessel. Note that this are dependent on the sail system angle with respect to the vessel body.
4. The sail system inertia is assumed to be neglectable with respect to the inertia of the vessel, hence the inertia of the sail system can either be neglected or used for a fixed sail system angle. Making the system inertia matrix independent of the system states. This will be valid from a design where the sail system is considerably lighter than the vessel. This is also desirable to keep CG as low as possible.
5. Sail system aerodynamic center (AC) is located at the sail system pivot point, which implies that the lift and drag forces produced by the sail will not cause any moment around the sail system pivot point and the heading angle of the sail system is hence independent of forces induced by the sail foil. This allows the sail system heading to be controlled by the tail angle together with the direction of the relative velocity with respect to the wind.
6. Lift and drag due to the tail is small compared to the sail, and mainly contribute to controlling the heading of the sail system relative to the wind. These are hence neglected in the forces from the sail system to the vessel.
7. The roll and pitch angles, together with their respective rates, are small and do not affect the sail system.

Vessel

1. The sail system and keel is assumed to be mounted symmetric on the xy-center of the vessel (on the center line through the CG) in order to avoid/reduce the sail system and keel-induced moment around the vessel z-axis.
2. The mass and inertia of the rudder are assumed to be neglectable.

Note that any sail system design where the tail and sail AC are "behind" the pivot point will align with the relative velocity with respect to wind, when the tail angle is zero. This is also true when the AC of the sail is located at the pivot point. Any perturbation from the alignment will cause a non-zero tail AOA, Where the resulting moment will try to bring the system towards zero AOA.

4.2.2 Sail system

Using the assumptions given in section 4.2.1, the sail system attitude dynamics can be decoupled from the rest of the system. Where the tail is used to control the angle of attack of the sail and the sail foil is used to induce forces that will be transferred to the vessel, making it the propulsion system of the saildrone.

Sail system attitude dynamics

As the sail system heading is independent of the vessel heading, it is useful to express the sail system heading with respect to NED with a separate state ξ . This state is related to the state ψ and ϵ through equation 4.17. For small angles in roll and pitch the relation in equation 4.18 is valid.

$$\dot{\xi} = r + \dot{\epsilon} \quad (4.17)$$

$$\xi = \psi + \epsilon \quad (4.18)$$

Considering the rotational dynamics of the sail system in $\{s\}$ the resulting diagonal inertia matrix around the Sail pivot point (SPP) can be described as:

$$I_{ss}^s = I_{cw}^s + I_s^s + I_t^s \quad (4.19)$$

Where the inertia of the counter weight (cw), sail (s) and tail (t) are given in $\{s\}$, which is located at the sail pivot point.

The sail frame $\{s\}$ is a rotating reference frame, hence forces and moment due to Coriolis and centripetal matrix will be present. However, these forces are functions of quadratic and product angular velocity, which will be small when the angular velocity is low. The roll and pitch dynamics of the saildrone are assumed to be relatively slow and the wind direction is assumed to be slowly varying. Hence the dominating dynamics of the sail system will be captured by the model even when these terms are omitted. Actually, for the sail system, the only degree of freedom relative to the vessel is the rotation around the z-axis. Using the fundamental equation 2.13, and extracting the expression related to the rotation around z, give equation 4.20. Now assuming that the sails rotation rates $\omega_{ns,x}^s$ and $\omega_{ns,y}^s$ are small, equation 4.21 is obtained.

$$I_{ss,zz}^s \ddot{\xi} + (I_{ss,yy}^s - I_{ss,xx}^s) \omega_{ns,x}^s \omega_{ns,y}^s = m_s^s \quad (4.20)$$

$$I_{ss,zz}^s \ddot{\xi} = m_s^s \quad (4.21)$$

where $I_{ss,xx}^s$, $I_{ss,yy}^s$ and $I_{ss,zz}^s$ is the diagonal elements of I_{ss}^s .

Denoting the magnitude and angle of the relative velocity vector with respect to wind at the SPP, as U_{rw} and β_{rw} respectively. Which is the definition used in figure 4.4. The relative velocity of the tail in $\{s\}$ can be described as equation 4.22. The relative velocity in the tail frame is described in equation 4.23 which is $v_{r,t}^s$ rotated into the tail frame.

$$v_{r,t}^s = \begin{bmatrix} U_{rw} \cos(\beta_{rw} - \xi) \\ U_{rw} \sin(\beta_{rw} - \xi) + r_{stAC,x}^s \dot{\xi} \\ 0 \end{bmatrix} \quad (4.22)$$

$$v_{r,t}^t = \begin{bmatrix} U_{rw} \cos(-\alpha_s + \delta_t) + \dot{\xi} r_{stAC,x}^s \sin(\delta_t) \\ \dot{\xi} r_{stAC,x}^s \cos(\delta_t) - U_{rw} \sin(-\alpha_s + \delta_t) \\ 0 \end{bmatrix} \quad (4.23)$$

Note that $\beta_{rw} - \xi = \alpha_s$, the AOA of the sail. While $r_{stAC,x}$ is the x-component of vector from the SPP to tail AC in $\{s\}$.

Equation 4.23 together with the formulas in equation 4.24, 2.32 and 2.34 can then be utilized to calculate the lift force $F_{lift,t}$ given by the tail.

$$\alpha_t = \text{atan2}\left(\frac{v_{r,t,y}^t}{v_{r,t,x}^t}\right) \quad (4.24)$$

$$F_{lift,t} = k_{CL_t} \alpha_t (v_{r,t}^t)^T v_{r,t}^t \quad (4.25)$$

Due to the induced lift force of the tail, the tail AOA will converge to zero, meaning the sail AOA will converge to the tail angle as a result of the sail system design. This is not entirely true as at large sail AOA will contribute with a moment as well, however it is assumed to be accurate enough for moderate tail angles.

Exploiting this relation using a first order Tailor series expansion to linearize equation 4.25 around $[\alpha_s \ \dot{\xi}] = [\delta_t \ 0]$, result in the force expressed in equation 4.26. Which describes the lift force acting in the tail AC for small tail AOA. Notice that the sign of the lift force is changed when expressed in the tail frame. The resulting moment acting on the sail system can be described by equation 4.27 which is obtained by rotating the force into the sail frame and translating it to the SPP. It can be observed that the effective moment arm is reduced when the sail AOA increases.

$$F_{lift}^t = -U_{rw} k_{CL_t} r_{stAC,x}^s \cos(\beta_{rw} - \xi) \dot{\xi} - U_{rw}^2 k_{CL_t} (\beta_{rw} - \xi) + U_{rw}^2 k_{CL_t} \delta_t \quad (4.26)$$

$$m_s^s = r_{stAC,x}^s \cos(\beta_{rw} - \xi) F_{lift,y}^t \quad (4.27)$$

Using the moment, the sail system attitude dynamics can be described as equation 4.28, where the state space form is given in equation 4.29. Remember that $\beta_{rw} - \xi = \alpha_s$, and that β_{rw} can be assumed to be constant or slowly varying, implying that $\dot{\alpha}_s \approx -\dot{\xi}$. Which imply that the system can be rewritten in terms of α_s which is useful for angle of attack controller design.

$$I_{ss,zz}^s \ddot{\xi} = -K_a U_{rw} \dot{\xi} - K_b U_{rw}^2 (\beta_{rw} - \xi) + K_b U_{rw}^2 \delta_t \quad (4.28a)$$

$$K_a = k_{CL_t} r_{stAC,x}^s \cos^2(\beta_{rw} - \xi) \quad (4.28b)$$

$$K_b = k_{CL_t} r_{stAC,x}^s \cos(\beta_{rw} - \xi) \quad (4.28c)$$

$$\begin{bmatrix} \dot{\xi} \\ \ddot{\xi} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \frac{K_b U_{rw}^2}{I_{ss,zz}^s} & \frac{-K_a U_{rw}}{I_{ss,zz}^s} \end{bmatrix} \begin{bmatrix} \xi \\ \dot{\xi} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \frac{-K_b U_{rw}^2}{I_{ss,zz}^s} & \frac{K_b U_{rw}^2}{I_{ss,zz}^s} \end{bmatrix} \begin{bmatrix} \beta_{rw} \\ \delta_t \end{bmatrix} \quad (4.29)$$

The equations indicate that the sail heading angle will align with the direction of the sum of the relative velocity angle β_{rw} and the tail angle δ_t at steady state. Which is equivalent to $\alpha_s = \delta_t$.

Sail induced forces

By studying figure 4.7 together with the expression for the lift and drag forces in equation 2.30 and 2.29, a simplified expression for the lift and drag forces acting on the sail can be obtained for small sail AOA.

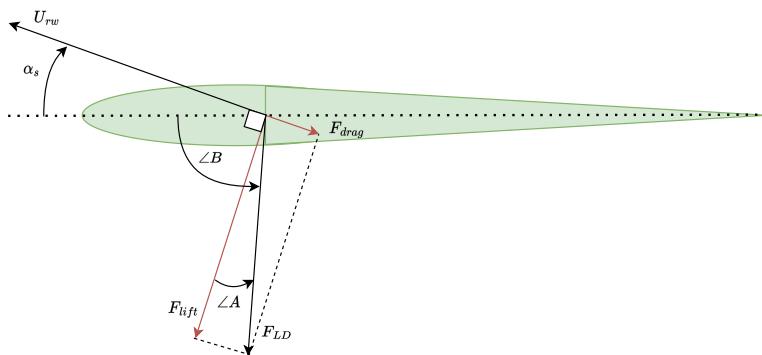


Figure 4.7: Sail forces

The angle $\angle A$ can be described as equation 4.30, which can be reduced to equation 4.31, when assuming that the wing is symmetric ($C_{L_0} = 0$) and the AOA dependent

drag dominate the parasitic drag. This will not be true for sail AOA $\rightarrow 0$, as seen from equation 2.38, as the only contributor becomes the parasitic drag. However, the magnitude of the drag and lift forces will be small and have a neglectable effect.

$$\angle A = -\tan^{-1}\left(\frac{F_{drag}}{F_{lift}}\right) = -\tan^{-1}\left(\frac{C_D(\alpha_s)}{C_L(\alpha_s)}\right) \quad (4.30)$$

$$\angle A \approx -\underbrace{\left(\frac{C_{L_\alpha}}{\pi e_{os} AR}\right)}_{\kappa} \alpha_s \quad (4.31)$$

Further, $\tan^{-1}()$ is linear close to zero, hence for small values of $\kappa\alpha_s$ the expression in equation 4.31 can be used. This also turn out to be true, for the specific design $\kappa \approx 0.3$ and α_s is assumed to be moderate.

The magnitude of the lift and drag force F_{LD} can be described as equation 4.32. Using the lift/drag ratio in equation 4.33, the equation can be rewritten as a function of lift force and the inverse of the lift/drag ratio. The lift/drag ratio is high on average, hence the square of the inverse will be very small and the magnitude can be assumed to be caused by the lift force alone, $F_{LD} \approx F_{lift}$.

$$F_{LD} = \sqrt{F_{lift}^2 + F_{drag}^2} \approx U_r^2 \alpha k_{CL} \sqrt{1 + \left(\frac{1}{C(\alpha)}\right)^2} \quad (4.32)$$

$$C(\alpha) = \frac{C_L(\alpha)}{C_D(\alpha)} \quad (4.33)$$

$$F_{LD} \approx F_{lift} \quad (4.34)$$

Using figure 4.7 together with equation 4.31 and 4.34, one can describe the resulting lift-drag vector in the sail frame by equation 4.35, where the angle $\angle B = (-\frac{\pi}{2} + \angle A + \alpha_s)$. Simplifying the expression using a small angle approximation give equation 4.36, which will be valid for a wide range of angle when $0 < (1 - \kappa) < 1$. The result is similar to the simplified rudder forces presented in section 4.2.5 with a additional foil design dependent constant correction term that reduces the resulting force along the sail frame x-axis. In addition, the sail AOA is used instead of the tail deflection angle.

$$f_s^s = F_{LD} \begin{bmatrix} \cos(\angle B) \\ \sin(\angle B) \end{bmatrix} = (U_{rw})^2 k_{CLs} \begin{bmatrix} \alpha_s \sin(\alpha_s(1 - \kappa)) \\ -\alpha_s \cos(\alpha_s(1 - \kappa)) \end{bmatrix} \quad (4.35)$$

$$f_s^s \approx (U_{rw})^2 k_{CLs} \begin{bmatrix} \alpha_s^2(1 - \kappa) \\ -\alpha_s \end{bmatrix} \quad (4.36)$$

The resulting 4 DOF generalized force vector, describing the forces transferred from the sail system to the vessel through the SPP, can be formulated as the fol-

lowing equation when neglecting the contribution from the tail. Where the transformation matrix is the reduced DOF $K_2^{\tau_1}$ transformation. The resulting sail system is illustrated in figure 4.8, where the magnitude U_{rw} and angle β_{rw} of the relative velocity can be found from the relative velocity vector v_{rw}^n .

$$\tau_{ss}^b(\epsilon, v_{rw}^n, \delta_t) = \begin{bmatrix} \cos(\epsilon) & -\sin(\epsilon) & 0 & 0 \\ \sin(\epsilon) & \cos(\epsilon) & 0 & 0 \\ -r_{bs,z}^b \sin(\epsilon) & -r_{bs,z}^b \cos(\epsilon) & \cos(\epsilon) & 0 \\ r_{bs,x}^b \sin(\epsilon) & r_{bs,x}^b \cos(\epsilon) & 0 & 1 \end{bmatrix} \begin{bmatrix} f_{s,x}^s \\ f_{s,y}^s \\ 0 \\ 0 \end{bmatrix} \quad (4.37)$$

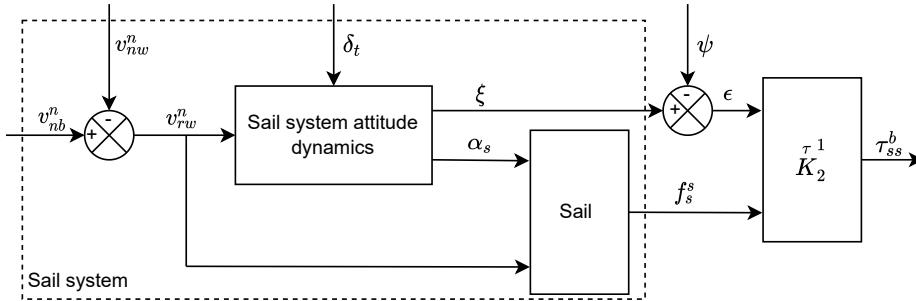


Figure 4.8: Sail system block diagram

4.2.3 Simplified rudder and tail model

The simplified tail and rudder model are specified as first-order systems with time constant T_t and T_r . This is identical to the rudder model presented in section 4.1.7. Further details are outlined there.

4.2.4 4 DOF Vessel model

A 4 DOF nonlinear vessel model can be obtained from the matrices and the model presented in section 3.2. Where the model equation can be rewritten as equation 4.38, where the $N(v_{rc})$ matrix is the collection of the terms $C(v_{rc})v_{rc}$, Dv_{rc} and $D_n(v_{rc})v_{rc}$.

$$M \ddot{v}_{rc} + N(v_{rc})v_{rc} + G\eta = \tau \quad (4.38)$$

A 4 DOF model is used under the assumption that the pitch angle and heave motion are small and neglectable. Which will be valid for maneuvering at calm sea or moderate waves, as the system has high stiffness (GM_L) in pitch and the heave motion will be small for moderate waves.

The system inertia matrix can be written as equation 4.40 under assumption that the saildrone is starboard-port ($x^b z^b$ -plane) symmetric and the added mass matrix is diagonal. To further simplify the model, the distance from CO to CG is assumed to be small, resulting in a diagonal M matrix.

$${}^4M = \begin{bmatrix} m - X_{\dot{u}} & 0 & 0 & 0 \\ 0 & m - Y_{\dot{v}} & -m r_{bg,z}^b & m r_{bg,x}^b \\ 0 & -m r_{bg,z}^b & I_x - K_p & -I_{xz} \\ 0 & m r_{bg,x}^b & -I_{zx} & I_z - N_r \end{bmatrix} \quad (4.39)$$

$${}^4M \approx diag([m - X_{\dot{u}} \ m - Y_{\dot{v}} \ I_x - K_p \ I_z - N_r]) \quad (4.40)$$

Notice that the system inertia matrix here is independent of the system states, as opposed to the one used in the simulation model. This is imposed by the assumptions that the inertia of the sail system is neglectable and the sail system CG is located at the SPP. The inertia of the sail system is however included at a fixed angle ($\epsilon = 0$), but this could also be omitted.

The reduced matrix 4N can be written as equation 4.41, by extracting the elements from the 6 DOF damping and Coriolis-centripetal matrices. Using the same assumption as for the system inertia matrix, with the vector from CO to CG being small and the fact that the yaw-rate will be limited. The matrix can be simplified to equation 4.42.

$${}^4N(v_{rc}) = \begin{bmatrix} -X_u - X_{nsd}^* |u_{rc}| & -mr & mr r_{bgz}^b & -mr r_{bgx}^b + Y_{\dot{v}} v_{rc} \\ mr & -Y_v & 0 & -X_{\dot{u}} u_{rc} \\ -mr r_{bgz}^b & 0 & -K_p & 0 \\ mr r_{bgx}^b - Y_{\dot{v}} v_{rc} & X_{\dot{u}} u_{rc} & 0 & -N_r \end{bmatrix} \quad (4.41)$$

$${}^4N(v_{rc}) = \begin{bmatrix} -X_u - X_{nsd}^* |u_{rc}| & -mr & 0 & Y_{\dot{v}} v_{rc} \\ mr & -Y_v & 0 & -X_{\dot{u}} u_{rc} \\ 0 & 0 & -K_p & 0 \\ -Y_{\dot{v}} v_{rc} & X_{\dot{u}} u_{rc} & 0 & -N_r \end{bmatrix} \quad (4.42)$$

Where the cross-flow drag terms are removed due to there complexity. In addition a simplified nonlinear surge damping X_{nsd}^* described by equation 4.43 is used, where $C_f(u_{rc})$ from the original expression in equation 3.39 is selected at constant velocity u_{rc}^* . Note that the nonlinear cross-flow drag terms are not corrected for, as these are often small terms compared to the surge damping during transit as $u_{rc}^2 \gg v_{rc}^2$ and $u_{rc}^2 \gg r^2$.

$$X_{nsd}^* = -\frac{1}{2}\rho S_h(1 + k_f)C_f(u_{rc}^*) \quad (4.43)$$

The 4 DOF restoring forces matrix can written as equation 4.44 using the coefficients from equation 3.44. Here the restoring forces only contribute in roll.

$${}^4G = \text{diag}([0 \ 0 \ GM_T mg \ 0]) \quad (4.44)$$

4.2.5 Rudder contribution

The rudder forces can be implemented with the rudder force equations propose on page 248 in reference [16]. Where only the forces caused by the rudder deflection is used, as seen in equation 4.45 and 4.46.

$$F_{r,x}^b = -k_{CL_r} U_{rc}^2 \delta_r^2 \quad (4.45)$$

$$F_{r,y}^b = k_{CL_r} U_{rc}^2 \delta_r \quad (4.46)$$

The proposed force equation 4.46 and the resulting moment was observed to create straight-line unstable saildrone due to the dominance of the destabilizing Munk moment ($(X_{\dot{u}} - Y_{\dot{v}}) u_{rc} v_{rc}$). However, the simulation model from section 4.1 remained straight-line stable, which means that the craft continue on a straight line after being subjected to a disturbance in yaw. The proposed force equation and its resulting moment do not include nonlinear yaw-damping caused by the rudder and the relative velocity, which is a significant contribution to the deviation between the two model. As an alternative, an identical approach as used for the sail system attitude dynamics can be applied, as the influence of the tail and rudder are the same except in two different mediums. This approach was observed to better reflect the dynamics of the simulation model.

The derivation of the rudder forces is identical to the tail lift force described in section 4.2.2, hence only the final result is shown her. Rewriting equation 4.26 in terms of U_{rc} , ψ , k_{CL_r} , β and rotating it into {b} give the rudder force along the body y-axis expressed in equation 4.47. Where the side slip angle $\beta = \beta_{rc} - \psi$.

$$F_{r,y}^b = \cos(\delta_r)(-k_{CL_r} r_{brAC,x}^b U_{rc} \cos(\beta)\dot{\psi} - k_{CL_r} U_{rc}^2 \beta + k_{CL_r} U_{rc}^2 \delta_r) \quad (4.47)$$

Further, the expression can be simplified by assuming the sideslip β and rudder deflection δ_r are small:

$$F_{r,y}^b = -k_{CL_r} r_{brAC,x}^b U_{rc} \dot{\psi} - k_{CL_r} U_{rc}^2 \beta + k_{CL_r} U_{rc}^2 \delta_r \quad (4.48)$$

The resulting generalized force vector is then obtained by combining the position vector of the rudder and the force equation 4.48. The simplified force along the body x-axis is obtained by combining equation 4.45 and the parasitic drag component. Where it is assumed that the parasitic drag act along the body x-axis as $u_{rc} \gg v_{rc}$. The generalized force vector becomes:

$$\tau_r^b(\eta, v_{rc}, \delta_r) = \begin{bmatrix} F_{r,x}^b - \frac{\rho C_{Dp} S_r U_{rc}^2}{2} \\ F_{r,y}^b \\ -r_{brAC,z}^b F_{r,y}^b \\ r_{brAC,x}^b F_{r,y}^b \end{bmatrix} \quad (4.49)$$

$\tau_r^b(\eta, v_{rc}, \delta_r)$ both depend on the system input δ_t and the states of the saildrone. Writing out the equation and separating input vector ${}^c\tau_r^b(\delta_r)$ from the state dependent terms collected in ${}^s\tau_r^b(\eta, v_{rc})$, result in the following equation:

$$\tau_r^b(\eta, v_{rc}, \delta_r) = \underbrace{\begin{bmatrix} -\frac{\rho C_{Dp} S_r U_{rc}^2}{2} \\ -U_{rc} k_{CL_r} r_{brAC,x}^b \dot{\psi} - U_{rc}^2 k_{CL_r} \beta \\ r_{brAC,z}^b (U_{rc} k_{CL_r} r_{brAC,x}^b \dot{\psi} + U_{rc}^2 k_{CL_r} \beta) \\ -r_{brAC,x}^b (U_{rc} k_{CL_r} r_{brAC,x}^b \dot{\psi} + U_{rc}^2 k_{CL_r} \beta) \end{bmatrix}}_{{}^s\tau_r^b(\eta, v_{rc})} + \underbrace{\begin{bmatrix} -U_{rc}^2 k_{CL_r} \delta_r^2 \\ U_{rc}^2 k_{CL_r} \delta_r \\ -r_{brAC,z}^b U_{rc}^2 k_{CL_r} \delta_r \\ r_{brAC,x}^b U_{rc}^2 k_{CL_r} \delta_r \end{bmatrix}}_{{}^c\tau_r^b(v_{rc}, \delta_r)} \quad (4.50)$$

The equation can further be expressed in terms of the relative sway velocity v_{rc} rather than the sideslip angle using the relation:

$$\beta = \sin^{-1}\left(\frac{v_{rc}}{U_{rc}}\right) \stackrel{\beta \text{ small}}{\Rightarrow} \beta \approx \frac{v_{rc}}{U_{rc}} \quad (4.51)$$

Utilizing equation 4.51 together with 4.50 the state dependent terms can be written similar to the N matrix. Note that the assumption $u_{rc} \gg v_{rc}$ imply that $U_{rc} \approx u_{rc}$.

$${}^s\tau_r^b(v_{rc}) = \underbrace{\begin{bmatrix} -\frac{\rho C_{Dp} S_r |u_{rc}|}{2} & 0 & 0 & 0 \\ 0 & -u_{rc} k_{CL_r} & 0 & -u_{rc} k_{CL_r} r_{brAC,x}^b \\ 0 & r_{brAC,z}^b u_{rc} k_{CL_r} & 0 & r_{brAC,z}^b u_{rc} k_{CL_r} r_{brAC,x}^b \\ 0 & -r_{brAC,x}^b u_{rc} k_{CL_r} & 0 & -(r_{brAC,x}^b)^2 u_{rc} k_{CL_r} \end{bmatrix}}_{N^R(v_{rc})} v_{rc} \quad (4.52)$$

4.2.6 Keel contribution

The keel contributes in the same way as the sail, where lift is dominating. The forces induced by the keel acting in the body frame can hence be described with the same equations as the sail induced forces in the sail frame. Rewriting equation 4.36 with respect to the keel give the following equation.

$$f_k^b \approx (U_{rc})^2 k_{CL_k} \begin{bmatrix} \alpha_k^2(1-\kappa_k) \\ -\alpha_k \end{bmatrix} \quad (4.53)$$

Assuming the keel AC is located at the xy-center of the vessel and the relative velocity due to the yaw-rate is neglectable. The keel AOA can be written as equation 4.54. By assuming a small AOA the expression can be reduced even further.

$$\alpha_k = \sin^{-1} \left(\frac{\nu_{rc} - r_{bkAC,z}^b p}{U_{rc}} \right) \xrightarrow{\alpha_k \text{ small}} \alpha_k \approx \frac{\nu_{rc} - r_{bkAC,z}^b p}{U_{rc}} \quad (4.54)$$

The keel forces then becomes:

$$F_{k,x}^b = k_{CL_k} (\nu_{rc}^2 - 2\nu_{rc} r_{bkAC,z}^b p + (r_{bkAC,z}^b p)^2)(1-\kappa_k) \quad (4.55)$$

$$F_{k,y}^b = -k_{CL_k} U_{rc} (\nu_{rc} - r_{bkAC,z}^b p) \quad (4.56)$$

The generalized force vector can be written:

$$\tau_k^b(\nu_{rc}) = \begin{bmatrix} F_{k,x}^b - \frac{\rho C_{Dp} S_k U_{rc}^2}{2} \\ F_{k,y}^b \\ -r_{bkAC,z}^b F_{k,y}^b \\ 0 \end{bmatrix} \quad (4.57)$$

The parasitic drag will be considerably larger in water than air. Hence it is included as well, in the same way as for the rudder forces. It is added directly to the force along the body x-axis under the assumption that $u_{rc} \gg \nu_{rc}$.

Rewriting equation 4.57 and using $U_{rc} \approx u_{rc}$ give the expression:

$$\tau_k^b(\nu_{rc}) = \underbrace{\begin{bmatrix} -\frac{\rho C_{Dp} S_k |u_{rc}|}{2} & N^K(1,2) & N^K(1,3) & 0 \\ 0 & -u_{rc} k_{CL_k} & u_{rc} k_{CL_k} r_{bkAC,z}^b & 0 \\ 0 & r_{bkAC,z}^b u_{rc} k_{CL_k} & -(r_{bkAC,z}^b)^2 u_{rc} k_{CL_k} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}}_{N^K(\nu_{rc})} \nu_{rc} \quad (4.58a)$$

$$N^K(1,2) = k_{CL_k} (\nu_{rc} - p r_{bkAC,z}^b)(1-\kappa_k) \quad (4.58b)$$

$$N^K(1,3) = k_{CL_k} (-\nu_{rc} r_{bkAC,z}^b + p(r_{bkAC,z}^b)^2)(1-\kappa_k) \quad (4.58c)$$

4.2.7 Saildrone model

Combining section 4.2.2 to 4.2.6 give the simplified saildrone model, which can be summarized as equation 4.59. Rearranging and gathering terms result in equation 4.60.

$$\begin{aligned} {}^4M\dot{\nu}_{rc} = & -{}^4N(\nu_{rc})\nu_{rc} - {}^4G\eta + \tau_k^b(\nu_{rc}) \\ & + {}^s\tau_r^b(\eta, \nu_{rc}) + {}^c\tau_r^b(\nu_{rc}, \delta_r) + \tau_{ss}^b(\epsilon, \nu_{rw}, \delta_t) \end{aligned} \quad (4.59)$$

$$\begin{aligned} {}^4M\dot{\nu}_{rc} = & (-{}^4N(\nu_{rc}) + N^K(\nu_{rc}) + N^R(\nu_{rc}))\nu_{rc} \\ & - {}^4G\eta + {}^c\tau_r^b(\nu_{rc}, \delta_r) + \tau_{ss}^b(\epsilon, \nu_{rw}, \delta_t) \end{aligned} \quad (4.60)$$

The sail system and the vessel can be viewed as symmetric problems. This is particularly evident when roll is neglected. Assuming roll and roll rate are small and neglectable, leading to the reduced 3 DOF system in equation 4.61, where one can observe this interesting perspective. The sail system and the vessel system are eventually the same, where the hull and keel together can be viewed as a sophisticated sail and the rudder's main objective is to control the heading of the vessel similar to the tail of the sail system. The forces created by the vessel and the sail system meet at the SSP and together create the resulting acceleration of the total mass. The simple structure is represented in figure 4.9. Where the rudder and tail angle is given by their respective first-order model.

$$\begin{aligned} {}^3M\dot{\nu}_{rc} = & \underbrace{(-{}^3N(\nu_{rc}) + {}^3N^K(\nu_{rc}) + {}^3N^R(\nu_{rc}))\nu_{rc} + {}^c\tau_r^b(\nu_{rc}, \delta_r)}_{\tau_{vessel}^b(\nu_{rc}, \delta_r)} + \tau_{ss}^b(\epsilon, \nu_{rw}, \delta_t) \end{aligned} \quad (4.61)$$

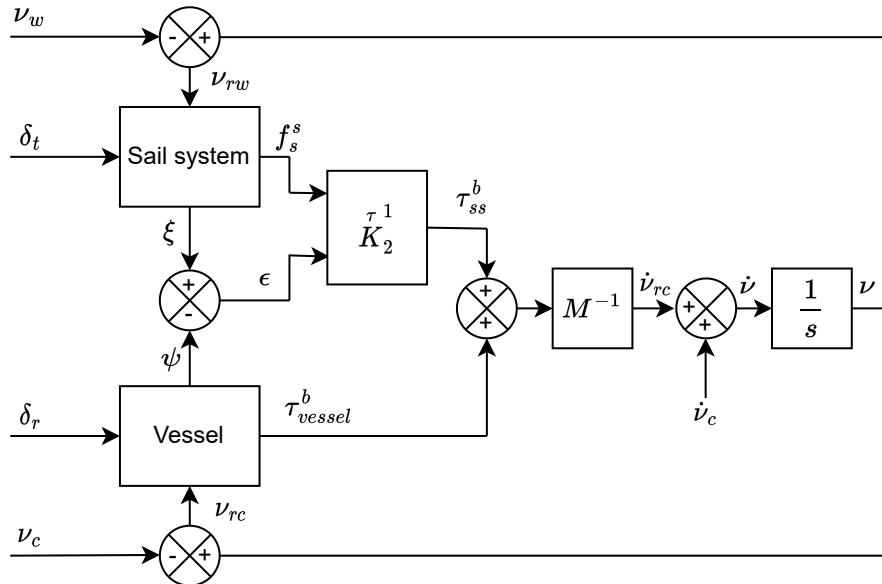


Figure 4.9: Simplified model block diagram

4.3 Model comparison and controllability

4.3.1 Model comparison

Using the system parameters given in the end of this chapter, the simulation model and the simplified model are compared to verify if the simplified model is representative.

In figure 4.10 the open loop response of the simulation model and the simplified model are compared. Where figure 4.10a shown the pose of the saildrone with 12 m/s wind from north, where the rudder is commanded to an angle of 15° for 5 seconds in both directions with 5 seconds apart. Here the sail controller tries to hold the sail AOA at 15° through the whole test. It can be observed that the two systems behave similarly both with respect to the heading and the resulting path. On the straight segment, the velocities are identical, indicating that the propulsion given by the sail matches well. This will be true as long as the sail AOA is kept under stall, as beyond this point, the lift will increase linearly in the simplified model while it drops in the simulation model. Beyond this point, an alternative model of the sail-induced forces needs to be used in the simplified model. During the turns, the simplified model tends to be more conservative and losses more velocity. Note that both models are straight-line stable. Despite the velocity deviation during the turn, it can be concluded that the dominating dynamics of the two systems are similar and the simplified model is representative.

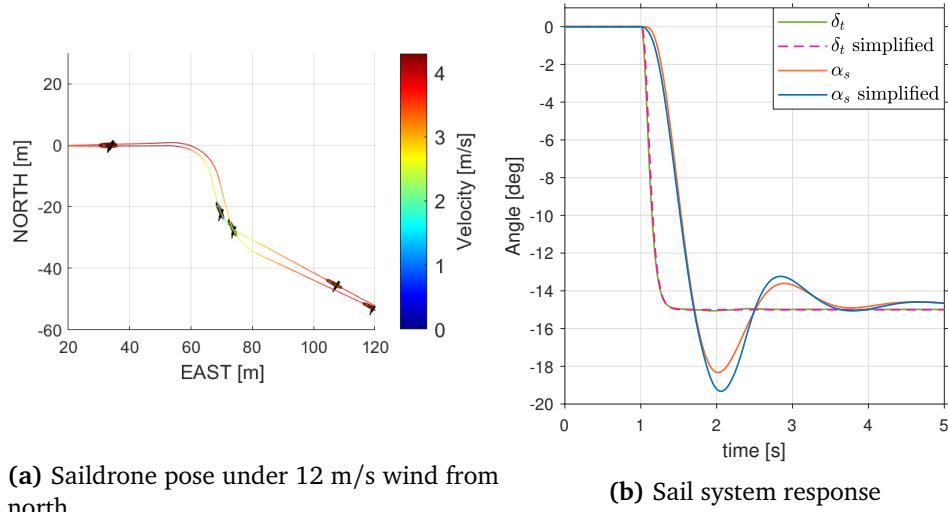


Figure 4.10: Open loop comparison of simulation- and simplified model

The open loop sail system response to a tail angle change of 15° is shown in figure 4.10b. Where the tail angle δ_t of the two models are identical, meaning the first

order model in the simplified model is a good representation of the tail angle dynamics together with the tail angle controller of the simulation model. From the figure it can also be seen that the transient sail AOA response of the two models are alike, however the amplitude of the simplified model is greater. A contributing factor to the deviation is drag of the tail, which is neglected in the simplified model.

Both of these open loop responses indicate that the simplified model captures the dominating dynamics of the saildrone. Meaning the simplified model can be applied for model based controller design. When using a simplified model it is however important to have sufficient margins as the control system will be subjected to disturbance from couplings and unmodeled dynamics when applied in the simulation model.

4.3.2 Controllability

As seen from the derivations of both the simplified and the simulation model, all system inputs are foil-induced forces depending on the relative velocities the respective subsystem sees. This implies that the systems are not controllable when the relative velocities are zero. Hence to the saildrone have to maintain velocity in order to stay controllable. This is the reason why jibing upwind is required when the initial velocity is too low, as the exit velocity of a tacking maneuver would make the saildrone end up being uncontrollable. For a formal discussion on sail craft controllability, see reference [32], where controllability is also discussed in the context of the dead zone. Where it is concluded that the system needs to have sufficient energy to maintain heading/course controllability during a tacking maneuver.

4.4 Model parameters approximation

The model parameters used in the simulations are given by the saildrone design shown in figure 2.1, which is designed with inspiration from the Explorer saildrone delivered by Saildrone Inc. As this is a theoretical model and no data is available for system identification the parameter estimates will be rough, but assumed to be accurate enough for simulation and controller design which are the scope of this thesis.

4.4.1 Saildrone inertia

The saildrone's elliptical shaped hull is assumed to have homogeneous mass distribution. The inertia with respect to the hull center is calculated using equation C.3.

The keel consists of 2 main components the ballast and the keel foil connecting the ballast to the hull. The ballast is modeled as a point mass while the keel foil is assumed to have a mass distribution like an elliptical cylinder around the z-axis going through the keel AC. The inertia of the keel foil can be calculated using equation C.1. While the inertia of the ballast can be calculated by using the parallel axis theorem in equation C.4.

The inertia of the sail and tail are both approximated by the inertia of a rectangular parallelepiped, reflecting the light outer foil skin and heavier inner structure. The inertia with respect to the mass center is given equation C.2. Similar to the ballast the sail system counterweight is modeled as a point mass and its inertia can be calculated the same way.

The inertias are then transformed to the relevant reference frames using the parallel axis theorem. Notice that for the simplified model the system inertia matrix will remain constant, however for the simulation model the inertia will change with the system states. Which also will be the case for the CG of the simulation model, if CG of the sail system is not located at the SPP.

4.4.2 Center of gravity and buoyancy

Both the center of gravity and buoyancy for a set of subsystems can be calculated as the weighted sum over all vectors to their respective *CG* or *CB*. The center of gravity is given in equation 4.62, where r_{bi}^b represent the vector from CO to the specific subsystem's *CG* with mass m_i . While the center of buoyancy is given by equation 4.63, where r_{bj}^b represent the vector from CO to the specific subsystem's *CB* with displaced water volume ∇_j .

$$r_{bg}^b = \frac{\sum_i r_{bi}^b m_i}{\sum_{i=1}^n m_i} \quad (4.62)$$

$$r_{bbou}^b = \frac{\sum_j r_{bj}^b \nabla_j}{\sum_{j=1}^n \nabla_j} \quad (4.63)$$

4.4.3 Hydrodynamic and hydrostatic parameters

The hydrodynamic and hydrostatic parameters are found using the equations given in section 3.2 together with the parameter estimates given in this section. The added mass matrix, natural frequencies and linear damping matrix are approximated using the rigid body inertia matrix and restoring coefficients matrix. Due to the different modeling approaches, the rigid body- and restoring coefficients matrices will differ, hence common hydrodynamic matrices for all models need to be established. To determine these, the parameters from the simplified

saildrone model is used, where the mass and inertia of all subsystems are included.

The hydrostatic forces will on the other hand, be different between the models, as the metacentric height will be higher when evaluating the vessel link alone (without the sail system) compared to the full system in the simplified model (including all subsystems). However, for the complete open chain model the actual metacentric height will be similar to the simplified model, as the moment due to gravity acting on the sail system will work against the restoring forces.

Added mass matrix

The added mass matrix M_A is assumed to be diagonal and is parameterized with respect to the system inertia matrix M following the best practice relationship found in the Maritime Robotics otter simulation model in MSS toolbox (available at <https://github.com/cybergalactic/MSS/blob/master/VESSELS/otter.m>). This is a rough estimation, however it is assumed to be accurate enough to obtain a representative simulation. Note that the developed simulator allows the user to change this data, hence empirical data can also be used for more accurate simulation. A detailed study of the hydrodynamic parameters and estimation of them is out of scope for this thesis. The relation used are as follows:

$$M_A = -\text{diag}([X_{\dot{u}}, Y_{\dot{v}}, Z_{\dot{w}}, K_{\dot{p}}, M_{\dot{q}}, N_{\dot{r}}]) \quad (4.64a)$$

$$X_{\dot{u}} = -\frac{2.7\rho\nabla^{5/3}}{l_{hull}^2} \quad (4.64b)$$

$$Y_{\dot{v}} = -1.5m \quad (4.64c)$$

$$Z_{\dot{w}} = -1.0m \quad (4.64d)$$

$$K_{\dot{p}} = -0.2I_{xx}^{CG} \quad (4.64e)$$

$$M_{\dot{q}} = -0.8I_{yy}^{CG} \quad (4.64f)$$

$$N_{\dot{r}} = -1.7I_{zz}^{CG} \quad (4.64g)$$

Where the added surge mass $X_{\dot{u}}$ is the formulation proposed in reference [33]. m is the total mass of the saildrone and I^{CG} is its inertia around CG.

Damping

The linear damping matrix is assumed to be diagonal which is common practice [16]. This allows the elements of the matrix to be determined by time constants and damping ratios shown in section 3.2.2. The natural frequency of the mass-damper-spring systems; heave, roll and pitch are calculated using the stiffness given by the restoring forces and masses given by the diagonal elements of the system inertia matrix (including added mass):

$$\omega_i = \sqrt{\frac{G_{ii}}{M_{ii}}} \quad i \in [3, 4, 5] \quad (4.65)$$

Where G_{ii} is the diagonal elements of the restoring coefficients matrix.

The nonlinear surge damping in equation 3.39 depend on the wetted surface of the hull S_h . Assuming the draft of the hull T is close to half of hull height, the wetted surface area is given by equation 4.66, where l_h and w_h are the length and width of the hull. The equation utilizes Knud Thomsen approximation for the surface area of an ellipsoid, which is adapted from reference [34][35].

$$S_h \approx 2\pi \left(\frac{(l_h w_h / 4)^{k_{KTF}} + (l_h T / 2)^{k_{KTF}} + (w_h T / 2)^{k_{KTF}}}{3} \right)^{1/k_{KTF}} \quad (4.66a)$$

$$k_{KTF} = 1.6075 \quad (4.66b)$$

The cross-flow drag integrals in equation 3.40-3.41 are discretized by summing over a set of 10 cross sections along the hull length. Where the cross-flow coefficients C_d^{2D} are calculated prior to simulation using the Hoerner curve in [16]. Where Hoerner gives the coefficients as a function of the draft $T(x)$ and beam $B(x)$ (hull width) along the hull length. The draft and beam along the hull are calculated using the ellipse equation 4.67, assuming $T \approx h_h/2$. Where a and b denote the ellipse radius along the major and minor axis, for the hull projected into the xy- and xz- plane.

$$B(x), T(x) = \sqrt{a^2 \left(1 - \left(\frac{x^2}{b^2} \right) \right)} \quad (4.67)$$

Draft and water plane area

The draft of the hull at calm sea is estimated by using the elliptic shape together with a correction coefficient C_{b_h} to adjust the volume of the hull approximation with respect to the psychical hull. Again the approximation is valid when $T \approx h_h/2$. In addition, the volume of the keel V_{keel} needs to be accounted for:

$$\nabla = \frac{m}{\rho} \quad (4.68)$$

$$\nabla_h = \nabla - V_{keel} \quad (4.69)$$

$$T = \frac{3\nabla_h}{2\pi \frac{w_h l_h}{4} C_{b_h}} \quad (4.70)$$

The waterplane A_{wp} is assumed to be constant for small perturbations when the restoring forces are derived in reference [16]. Which will be true at calm sea and $T \approx h_h/2$. The water plane area and the second moments of area of the elliptic waterplane are given:

$$A_{wp} \approx \frac{\pi w_h l_h}{4} \quad (4.71)$$

$$I_L = \frac{\pi}{4} \left(\frac{l_h}{2} \right)^3 \left(\frac{w_h}{2} \right) \quad (4.72)$$

$$I_T = \frac{\pi}{4} \left(\frac{w_h}{2} \right)^3 \left(\frac{l_h}{2} \right) \quad (4.73)$$

Saildrone design parameters

The key design parameters of the saildrone used for simulation in this thesis are summarized in table 4.1 and 4.2.

Table 4.1: Saildrone dimensions and subsystem weight

Subsystem	Symbol	Description	Value	Unit
Hull	h_{hull}	Height	0.7	m
	l_{hull}	Length	7	m
	w_{hull}	Width	1	m
	m_{hull}	Mass	750	kg
Keel	h_{Keel}	Height	2	m
	l_{Keel}	Length	0.6	m
	w_{Keel}	Width	0.07	m
	m_{Keel}	Mass	200	kg
Rudder	h_{Sail}	Height	0.8	m
	l_{Sail}	Length	0.4	m
	w_{Sail}	Width	0.12	m
Sail	h_{Sail}	Height	5	m
	l_{Sail}	Length	1	m
	w_{Sail}	Width	0.08	m
	m_{Sail}	Mass	40	kg
	m_{weight}	Counter weight mass	5	kg
Tail	h_{Sail}	Height	2.2	m
	l_{Sail}	Length	0.44	m
	w_{Sail}	Width	0.05	m
	m_{Sail}	Mass	5	kg
Saildrone	m	Total weight of saildrone	1000	kg

Table 4.2: Key position vectors

Symbol	Value	Unit
r_{bhCG}^b	$[0 \ 0 \ 0]^T$	m
r_{bkAC}^b	$[0 \ 0 \ 1.35]^T$	m
r_{bkCG}^b	$[0 \ 0 \ 2.15]^T$	m
r_{brAC}^b	$[-3.15 \ 0 \ 0.75]^T$	m
r_{bsAC}^b	$[0 \ 0 \ -2.85]^T$	m
r_{swCG}^s	$[2.30 \ 0 \ 0]^T$	m
r_{stAC}^s	$[-2.30 \ 0 \ 0]^T$	m

Where r_{swCG}^s is the vector from the sail frame origin to the center of gravity of the counter weight. For foils without specified CG, the CG and AC are located at the same point.

4.4.4 Foil coefficients

The simulation model of the entire saildrone needs a representative model of the foil-induced forces and moments at large angle of attack, as large AOA can occur both intentionally and unintentionally. Such situations can be caused by poor control strategy, system bugs, actuator failure, low velocity turns and more.

To represent the foil induced forces and moments from the sail, tail, keel and rudder, the foil theory presented in section 2.5 is applied together with the empiric functions found in reference [36]. In addition, the wind tunnel data from the NACA-0015 wing in appendix B is used as a reference when developing and tuning the models for lift, drag and moment coefficients.

Note that the wind tunnel data is given as section characteristics data which represent the coefficient of a foil of infinite aspect ratio [37]. This can be verified by comparing the initial slope of the section lift coefficient in figure B.1 and the slope given by equation 2.33 as AR goes to infinity, which will be $C_{L_a} = 2\pi$. This means that the slope in the linear range will in general be smaller than shown in figure B.1, due to finite aspect ratio of the real foil. It can also be observed that the linear slope will be an optimistic estimate of the lift coefficient close to the stall angle, however this is assumed to be sufficiently accurate for simulation.

In [36] an extensive modeling framework for finite aspect ratio foils is developed, where the models consist of a pre and post-stall region. Here a set of empirical functions based on a large set of wind tunnel data is proposed for detailed model-

ing. The paper also shows that, within the post-stall region, changes in aspect ratio mainly scale the coefficient curves while the shape remains similar. For this thesis, most details are left out to reduce the model's complexity while maintaining the dominating effects.

Lift coefficient

The lift model is represented by using the model proposed in equation 2.37, where the flat plate approximation is exchanged with equation 4.74, which better reflect the post stall lift. As the maximum lift of the flat plate model is at a higher angle of attack than what is given by figure B.1. $C_{L_{2max}}$ is the maximum post stall lift in the new model, which can be found using the empirical equation 4.75 according to [36].

$$C_{L_{fp}}^{new} = C_{L_{2max}} \sin(2\alpha) \quad (4.74)$$

$$C_{L_{2max}} = F_1(t/c)F_2(AR) \quad (4.75a)$$

$$F_1(t/c) = 1.190(1 - (t/c)^2) \quad (4.75b)$$

$$F_2(AR) = 0.65 + 0.35e^{-(9.0/AR)^{2.3}} \quad (4.75c)$$

Where c is the chord length, t are the foil thickness and AR is the finite aspect ratio. For the reference NACA-0015 the thickness to chord length ratio (t/c) will be 0.15.

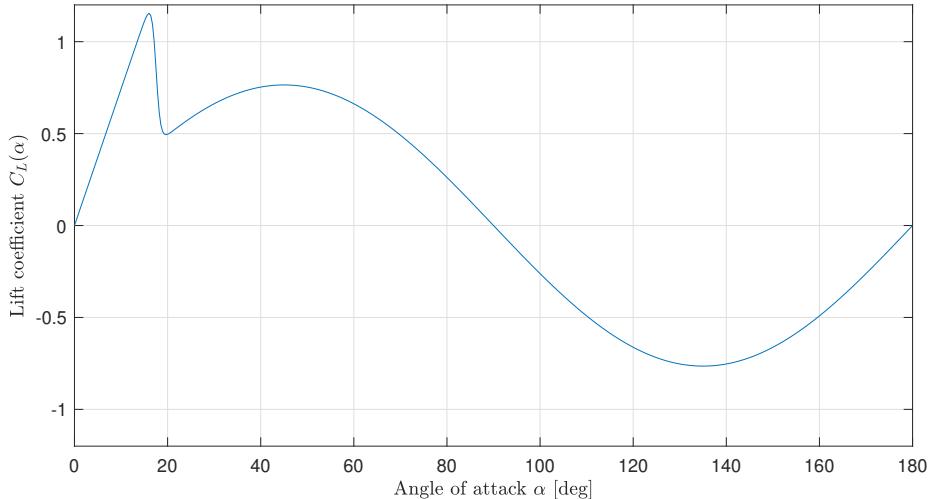


Figure 4.11: Lift coefficient approximation

Using equation 4.74 in the nonlinear lift model 2.37 give the result shown in figure 4.11. Where a transition gain $J = 120$ and cutoff angle $\alpha_0 = 17.5^\circ$ is utilized.

These parameters are used for all foils, while the aspect ratio varies with the foil design. The figure show the coefficients for the sail, with $(t/c) = 0.15$ and aspect ratio $AR = 5$. The scaled shape of the coefficient model correlates well with the NACA-0015 data up to AOA 90 degrees. The remaining also correlates well, but here the model doesn't reflect the stall angle and lift drop when reversing (AOA>90), however the result is assumed to be sufficiently accurate for simulation. As forward motion is essential, and there is no intention of reversing.

Drag coefficient

The quadratic drag model presented in section 2.5 is limited to moderate angles, hence the model in equation 4.76 is proposed as an better alternative. Which is inspired by the shape of the quadratic function and the drag coefficient data in figure B.2. The drag model is similar to equation 2.38, where α is replaced by $\sin(\alpha)$ and a additional term is added to set the peak drag using the gain $C_{D_{2max}}$. Which is found using empirical equation 4.77. Figure 4.12 show the drag coefficient model using the same parameters as in figure 4.11 and the parasitic drag $C_{D_p} = 0.008$.

$$C_D(\alpha) = C_{D_p} + \frac{(C_{L_0} + C_{L_a} \sin(\alpha))^2}{\pi e_{os} AR} + C_{D_{2max}} \sin^4(\alpha) \quad (4.76)$$

$$C_{D_{2max}} = G_1(t/c)G_2(AR) \quad (4.77a)$$

$$G_1(t/c) = 2.3e^{-(0.65(t/c))^{0.9}} \quad (4.77b)$$

$$G_2(AR) = 0.52 + 0.48e^{-(6.5/AR)^{1.1}} \quad (4.77c)$$

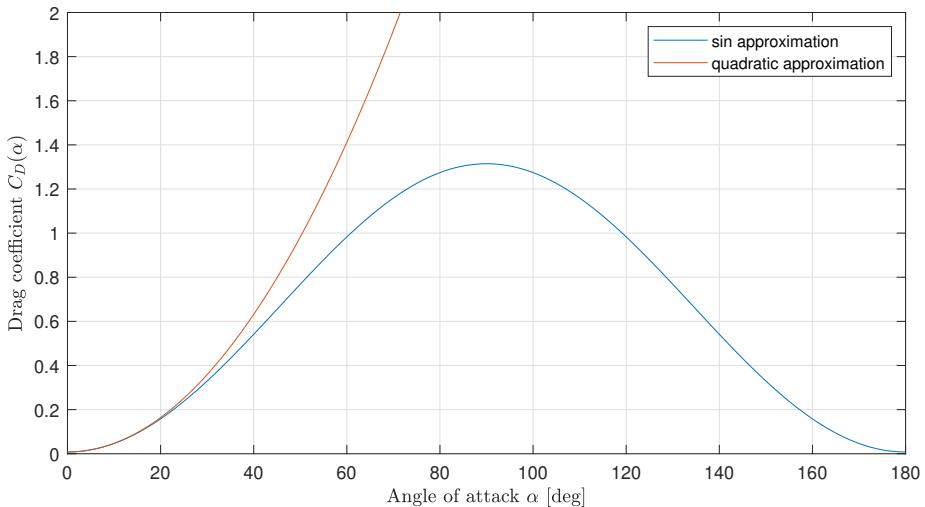


Figure 4.12: Drag coefficient approximation

Moment coefficient

Considering a symmetric foil and moderate angles of attack, the moment coefficient can be left out due to its low value. Under normal operation the rudder, keel, and tail will experience a moderate angle of attack, however, it might be of interest to maximize drag when sailing downwind, implying large sail AOA. Here the moment will be significant and counteract the moment created by the tail's lift, hence limiting the AOA of the sail.

The moment coefficient is approximated by equation 4.78, where the first linear term is used for small angles, while the nonlinear term dominates at large AOA. The transition is carried out by applying the blending function in equation 2.36. This will be a rough estimate as the data in figure B.3 has large spread. In addition, approximation models for the moment coefficient for the highly nonlinear post-stall region tend to be absent. The obtained model here is based on B.3 under the assumption that the foil cross-section and aspect ratio has less impact on the moment at the quarter chord when the flow has separated from the foil at angles beyond stall. Studying the wind tunnel data in reference [38] it can be observed that the moment coefficient decreases with decreasing aspect ratio, hence the model is tuned to be conservative. The less accurate model will not compromise the accuracy of the simulation model as it only contributes at significant AOA.

$$C_M(\alpha) = (1 - \sigma_L(\alpha))(C_{m_\alpha} \alpha) + \sigma_L(\alpha)k_{C_m} \sin(\alpha)\alpha \operatorname{sign}(\alpha) \quad (4.78)$$

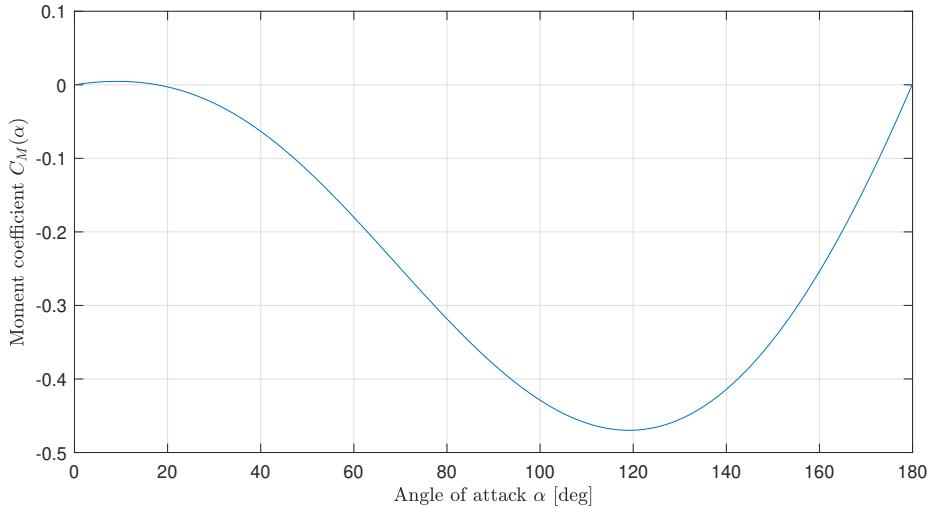


Figure 4.13: Moment coefficient approximation

Notice that the blending function's $\sigma_L(\alpha)$ parameters are not the same as for the lift coefficient. The applied parameters are the transition gain $J = 2$, cutoff angle

$\alpha_0 = 16^\circ$, linear slope $C_{m_\alpha} = 0.14$ and gain $k_{Cm} = 0.27$. The resulting approximation is shown in figure 4.13

Chapter 5

Simulation and motion control

This chapter presents an overview of the developed motion control system and the simulation framework used to evaluate its performance and working concept. The control and guidance systems that make up the motion control system are covered in detail in the coming chapters. Furthermore, the relation between the control system structure and the saildrone dynamics is discussed.

5.1 Simulator overview

The simulator utilized the simulation model developed in section 4.1 together with the system parameters established in section 4.4. While the simplified model from section 4.2 is available for comparison. The simulator also features a 2D visualization of the underway response of the saildrone. Both as animation and as discrete saildrone poses along the trajectory it takes. An example of the latter can be seen in figure 5.1, where the color of the trajectory reflects the velocity of the saildrone at the specific positions in $\{n\}$. The animation shows the actual pose of the saildrone as time evolves, which is useful both when debugging and verifying that the response is reasonable.

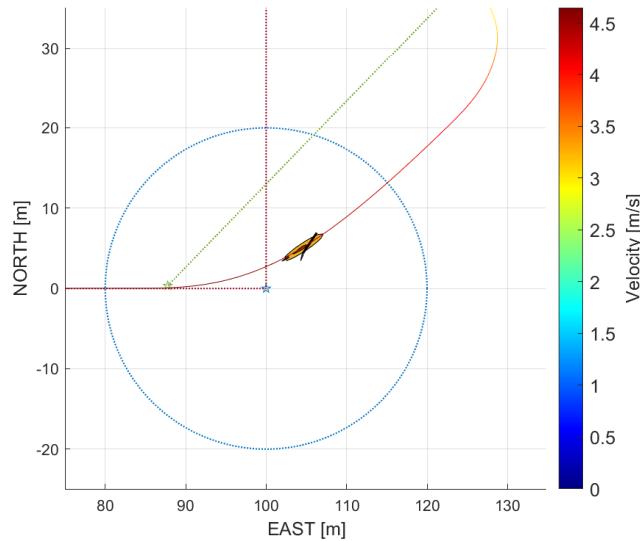


Figure 5.1: Simulation example

A description of the symbols used in the simulation plots are given in table 5.1. Notice that global and local waypoints (GWP and WP) are distinguished. Where WPs are local intermediate waypoints used to modify the path when the saildrone needs to deviate from the global path in order to reach the next GWP. The global path is defined by the straight line segments connecting the GWPs, which are the waypoints the saildrone are meant to visit. These can be provided by a user or a route optimization layer.

Table 5.1: Symbol and plot description

Description	Symbol
Global waypoint	★
Local waypoint	★
Path between global WPs
Path between local WPs
Reference frame x-axis	→
Reference frame y-axis	→
Circle of acceptance for global WPs	○
Obstacle	○

5.2 Guidance and control system overview

An overview of the controller and guidance system, saildrone dynamics and perception model is presented in figure 5.2. The control layer is developed through chapter 6, while the guidance system is presented in chapter 7. The overview also reflects the simulation setup.

The controller hierarchy reflects the structure of the simplified model, where the vessel and sail system are separated. Allowing the control problem to be divided into two separate sub-problems: control the heading/course of the vessel and control the heading/AOA of the sail, where the rudder and tail angles are controlled to achieve this. The separate control of the sail system is exploited to simplify the motion control problem by selecting an optimal sail AOA, based on the saildrone's heading and angle of the relative velocity β_{rw} . This implies that the guidance system does not need to provide a sail AOA command. Why heading is utilized instead of course is discussed in section 6.5.

By utilizing the solutions described above, the path following is reduced to controlling the course to reach the target. However, the problem is not trivial due to the constraints imposed by the sail propulsion. Hence a path following algorithm is used to decide how to best progress along the path, and control the desired course.

The top layer in the hierarchy, the path manager, is a simple waypoint switching mechanism, which updates the active waypoints (GWP) as the saildrone progresses. The GWP are assumed to be specified by a user or a route optimization layer, considering land, islands, and other large structures.

The collision avoidance algorithm is placed between the path manager and the path following algorithm to provide alternative waypoints when the original path can lead to a collision. Where the alternative path is adjusted as the saildrone progresses and the collision avoidance problem is formulated such that crossing is allowed under avoidance.

A perception of the environment surrounding the saildrone is required to enable collision avoidance. It is archived by using a binary occupancy grid map fixed in $\{n\}$ where measurements are obtained by a set of equal space finite length rays originating at the saildrone. They give the intersection points relative to the saildrone's position if they encounter an occupied cell. These measurements will be ideal with no uncertainty as a binary occupancy map is used. The interpretation represents processed measurements from a LiDAR, radar, seafloor maps, or others. The obtained measurements are used to create velocity obstacles, which are further exploited in the collision avoidance algorithm.

The different controllers and sub-modules in the control and guidance layers are discrete in order to represent an implementation on a real system, where system resources can be limited. Hence the different controllers and sub-modules run at different sampling times dependent on how fast dynamics they control.

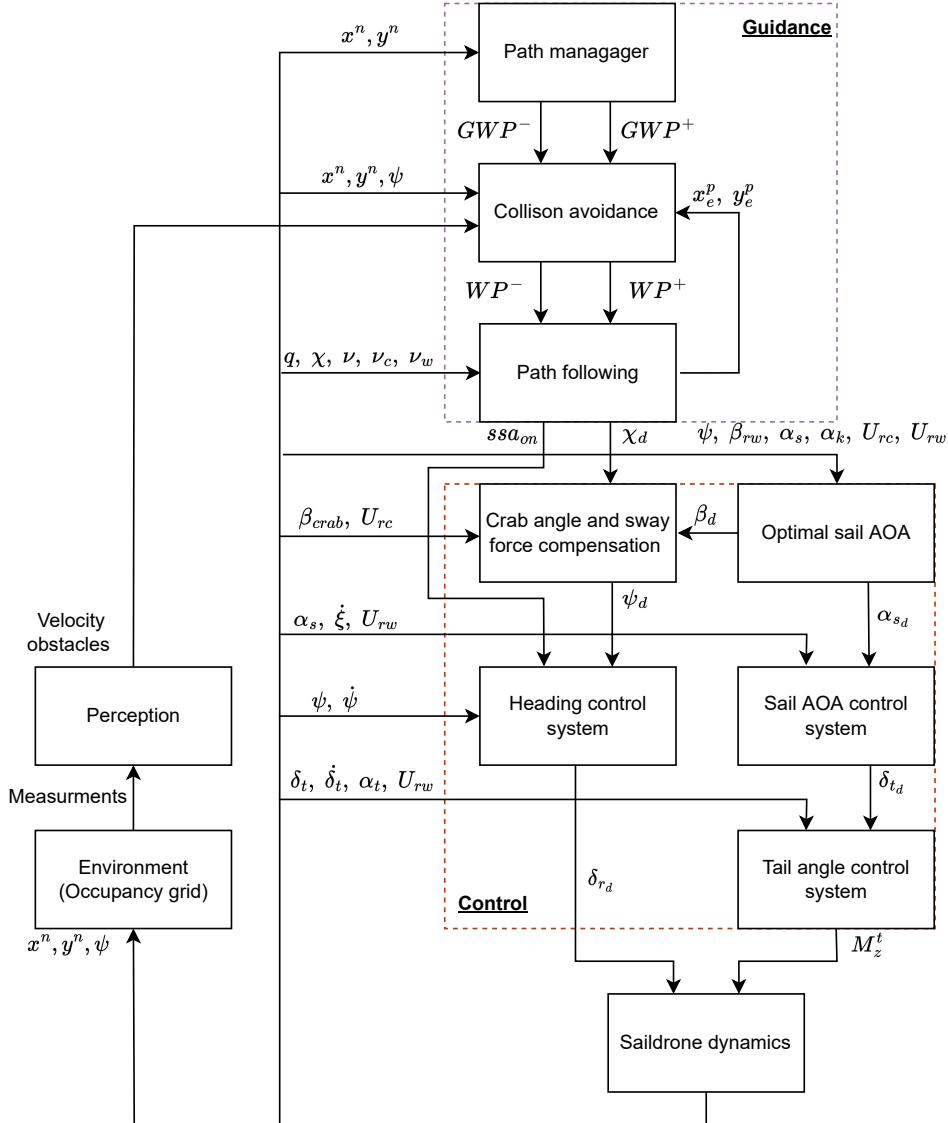


Figure 5.2: Guidance and control overview

5.2.1 Implementation

The simulation and motion control system are implemented using Matlab Simulink, where the simulator exploits Simulink's variable step solver. While the cre-

ation and measurements given by the binary occupancy grid map are functionality provided by Matlab. The MSS toolbox [16] (available at: <https://github.com/cybergalactic/MSS>) is exploited in transforms and other elementary operations.

The code for the entire implementation can be found at:
https://github.com/jorgen-benum/TTK4900_SAILDRONE

Control and sail propulsion

This chapter presented the utilized reference models, controllers, and offline optimization that maximizes the forward propulsion given by the sail. These together form the control system, which reflects the structure of the simplified model, where the control problem is divided into control of the saildrone's heading/-course and sail AOA.

A method is derived to exploit the lift and drag coefficients to make an offline sail AOA optimization that maximizes the forward propulsion. The underlying equations are also utilized to create a heading correction term to cancel the sail's sway force.

Further, the relation between the heading controller and course command is presented, along with a comparison of forward propulsion optimization with respect to heading and course.

At the end of the chapter, the tuning of the reference models and controllers are presented and discussed.

6.1 Reference model

Reference models are often used in motion control systems to create smooth and continuous reference signals to track from otherwise discontinuous changes in reference. Which can be jumps in reference velocity, attitude, position, or others.

Position and attitude reference models are often constructed by cascading a first-order low-pass filter and a mass-damper-spring system, resulting in third-order filtering of the reference signal r^n . The resulting state space formulation, containing multiple reference models, can be written:

$$\dot{x}_d = A_d x_d + B_d r^n \quad (6.1)$$

$$A_d = \begin{bmatrix} 0_{n \times n} & I_n & 0_{n \times n} \\ 0_{n \times n} & 0_{n \times n} & I_n \\ -\Omega^3 & -(2\zeta + I_n)\Omega^2 & -(2\zeta + I_n)\Omega \end{bmatrix}, \quad B_d = \begin{bmatrix} 0_{n \times n} \\ 0_{n \times n} \\ \Omega^3 \end{bmatrix} \quad (6.2)$$

Where $x_d = [\eta_d^T \quad \dot{\eta}_d^T \quad \ddot{\eta}_d^T]^T$ is the vector of smooth low-pass filtered references, that can be used in tracking control. While $\zeta = \zeta_{ref}$ and $\Omega = \omega_{n_{ref}}$ are the reference model's relative damping ratio and natural frequency respectively. The model formulation is adapted from [16] and the derivation details can be found there. Note that x_d remains smooth for a step in reference r^n . In addition the desired position/attitude η_d converges to r^n at steady state.

Saturation elements can be included to represent the system's velocity and acceleration limits to improve the reference model further. It will create references that are more likely feasible for the system dynamics. It is also important that the reference model is slower than the system dynamics for the tracking problem to be feasible. A critically damped system with a damping ratio of 1 is typically utilized to create a smooth reference without oscillations while keeping response time low.

6.2 Controllers

6.2.1 Tail angle controller

A tail actuator controller is needed to track the desired tail angle, where the controller outputs the actuator torque. Here a linear PD controller with a compensating feed forward term is utilized to track the reference signal.

The simplified model in equation 6.3 is used for tail angle controller design. The model is obtained by considering the decoupled sail system and assuming that the inertia of SAW is much greater than the inertia of the tail. Such that the sail system is approximately fixed in {n} from the perspective of the tail.

$$I_{t,zz}^t \ddot{\delta}_t = r_{ttAC,x}^t F_{tAC,y}^t + r_{ttCG,x}^t F_{tCG,y}^t + M_z^t \quad (6.3)$$

Where M_z^t is the actuator moment command.

For the sail system design considered in this thesis, the joint connecting the tail to the rest of the sail system has a small offset from the AC and CG. This offset implies that the aerodynamic and gravitational forces will cause a moment

$r_{ttAC,x}^t F_{tAC,y}^t + r_{ttCG,x}^t F_{tCG,y}^t$ that the controller must compensated for. The compensation can be done by integral action. However, some of the effects can be accounted for using feed forward.

Here the roll and pitch angle are assumed to be small such that the moment due to gravity is neglectable. In addition, the tail AOA is assumed to be small, implying that the moment contribution due to aerodynamics is caused by lift. This lead to the feed forward term shown in equation 6.4, where the term is reduced using the gain k_{red} to avoid overcompensation, as the model is significantly simplified.

$$M_z^t = -k_{red} r_{ttAC,x}^t F_{tAC,y}^t + \bar{M}_z^t \quad (6.4)$$

Introducing equation 6.4 into 6.3 result in a pure double integrator system:

$$\frac{\ddot{\delta}_t}{\bar{M}_z^t} = \frac{1}{I_{t,zz}^t} \quad (6.5)$$

The PD controller in equation 6.6 is applied to the system, using tracking control with the tail angle error denoted $\tilde{\delta}_t = \delta_t - \delta_{t_d}$. This result in error dynamics in equation 6.7. Which can be compared to the mass-damper-spring system in equation 6.8 to form a pole-placement algorithm for the controller gains.

$$\bar{M}_z^t = -K_p \tilde{\delta}_t - K_d \dot{\tilde{\delta}}_t + I_{t,zz}^t \ddot{\delta}_{t_d} \quad (6.6)$$

$$\ddot{\tilde{\delta}}_t + \frac{K_d}{I_{t,zz}^t} \dot{\tilde{\delta}}_t + \frac{K_p}{I_{t,zz}^t} \tilde{\delta}_t = 0 \quad (6.7)$$

$$\ddot{x} + 2\zeta \omega_n \dot{x} + \omega_n^2 x = 0 \quad (6.8)$$

The controller gains can be selected based on equation 6.9 - 6.11, which are adopted from reference [16]. Where the gains are given by the desired control bandwidth ω_b and the damping ratio ζ .

$$\omega_n = \frac{1}{\sqrt{1 - 2\zeta^2 + \sqrt{4\zeta^4 - 4\zeta^2 + 2}}} \omega_b \quad (6.9)$$

$$K_p = \omega_n^2 I_{t,zz}^t \quad (6.10)$$

$$K_d = 2\zeta \omega_n I_{t,zz}^t \quad (6.11)$$

Integral action is not included in this inner loop controller to avoid stability issues with successive integrators. In addition, the main concern is the outer loop error,

meaning a small inner loop error can be tolerated.

The tail angle tracking controller is shown in equation 6.12 with the feed forward term. Notice that the compensation term is significantly simplified; a more detailed term will yield better performance.

$$M_z^t = -k_{red} r_{ttAC,x}^t F_{tAC,y}^t - K_p \tilde{\delta}_t - K_d \dot{\tilde{\delta}}_t + I_{t,zz}^t \ddot{\tilde{\delta}}_{t_d} \quad (6.12)$$

Notice that the position of the joint relative to the tail AC is not optimal both from a controller perspective and power consumption as the controller will need to command a nonzero actuator moment to maintain a nonzero tail AOA. Moving the joint closer to the AC will reduce the effect. Another alternative is to use an actuator with brakes or self-locking ability. The original solution is, however, sufficient for simulation.

As the simulation model does not include any tail actuator model, the natural gain due to gear ratios is left out. This results in high controller gains as the controller gives the moment needed to control the tail angle directly.

6.2.2 Sail AOA controller

Without an actuator controlling the angle between the sail and the vessel, the sail system is underactuated, and its heading angle can only be controlled using the tail angle. This implies that the sail heading can only be adjusted within a limited range from the direction of the relative velocity β_{rw} . Hence it is convenient to control the sail AOA rather than the sail system heading itself.

The nonlinear velocity dependent sail system dynamics established in equation 4.28 can be exploited to form a velocity invariant linear system. Assuming the angle of the relative velocity vector β_{rw} is slowly varying such that equation 6.13 and 6.14 are valid. Then equation 4.28 can be rewritten in terms of α_s as shown in equation 6.15. In order to create the constant coefficients K_A and K_B , α_s^* is introduced as a constant angle, representing the angle the sail AOA should operate around.

$$\dot{\alpha}_s \approx -\dot{\xi} \quad (6.13)$$

$$\ddot{\alpha}_s \approx -\ddot{\xi} \quad (6.14)$$

$$I_{ss,zz}^s \ddot{\alpha}_s = \underbrace{-r_{stAC,x}^{s2} \cos^2(\alpha_s^*) k_{CL_t}}_{K_A} U_{rw} \dot{\alpha}_s + \underbrace{r_{stAC,x}^s \cos(\alpha_s^*) k_{CL_t}}_{K_B} U_{rw}^2 (\alpha_s - \delta_t) \quad (6.15)$$

Notice that the damping and stiffness of the second order system change with the operation point α_s^* , and the tail and sail system design parameters.

Velocity invariant linear system

Equation 6.15 has a particular form which can be transformed into a velocity invariant linear system as shown in reference [16] (appendix D). Where the normalized non-dimensional model can be used as a basis for gain scheduling, with U_{rw} as a measurement. Often allowing fast gain scheduling as velocity estimation/measurement rates are usually high.

Defining the non-dimensional variable Γ as in equation 6.16, where U is a velocity, L is a characteristic length and t represent time.

$$\Gamma = \frac{U}{L} t \quad (6.16)$$

Further, the first and second derivatives with respect to time can be written:

$$\frac{d}{dt} = \frac{d}{d\Gamma} \frac{d\Gamma}{dt} = \frac{d}{d\Gamma} \frac{U}{L} \quad (6.17)$$

$$\frac{d^2}{dt^2} = \frac{d}{d\Gamma} \frac{d\Gamma}{dt} \left(\frac{d}{d\Gamma} \frac{d\Gamma}{dt} \right) = \frac{d^2}{d\Gamma^2} \frac{U^2}{L^2} \quad (6.18)$$

Using the result above, with $U = U_{rw}$ and $L = 1$ meter, to transform the dynamics in equation 6.15:

$$I_{ss,zz}^s \frac{d^2\alpha_s}{d\Gamma^2} = -K_A \frac{d\alpha_s}{d\Gamma} + K_B(\alpha_s - \delta_t) \quad (6.19)$$

The resulting linear system can be controlled with the PID tracking controller in equation 6.20. Where the tracking error is denoted $\tilde{\alpha}_s = \alpha_s - \alpha_{s_d}$.

$$\delta_t = -\frac{I_{ss,zz}^s}{K_B} \frac{d^2\alpha_{s_d}}{d\Gamma^2} - \frac{K_A}{K_B} \frac{d\alpha_{s_d}}{d\Gamma} + \alpha_{s_d} - K_d \frac{d\tilde{\alpha}_s}{d\Gamma} - K_p \tilde{\alpha}_s - K_i \int \tilde{\alpha}_s d\Gamma \quad (6.20)$$

Introducing the controller into the system and using pole placement similar to section 6.2.1 give the following controller gains based on natural frequency and damping ratio:

$$K_p = \frac{\omega_n^2 I_{ss,zz}^s + K_B}{-K_B} \quad (6.21)$$

$$K_d = \frac{2\zeta\omega_n I_{ss,zz}^s - K_A}{-K_B} \quad (6.22)$$

$$K_i = \frac{\omega_n}{10} K_p \quad (6.23)$$

Here the natural frequency can be selected using the desired bandwidth together with equation 6.9. Notice here that K_B is negative as it includes the x-component of the vector from $\{s\}$ to the tail AC with respect to $\{s\}$.

The integral action gain is selected to create an integrator approximately 10 times slower than the natural frequency ω_n . The integrator will slightly change the system response, but by a neglectable amount when using reasonable integrator gain. Note that due to the natural stiffness of the system K_p can be negative for some desired natural frequency ω_n , meaning this must be checked to avoid negative integrator gain.

The tail angle can be saturated due to the restrictions the actuator and the joint impose. Hence the controller's integrator should be implemented with an anti-windup scheme.

Gain-scheduled controller

Transforming the controller back to the time domain results in the control law in equation 6.24. Here U_{rw} is the measurement of the relative velocity with respect to wind at the sail AC, which becomes the scheduling variable.

$$\delta_t = -\frac{I_{ss,zz}^s}{K_B U_{rw}^2} \ddot{\alpha}_{s_d} - \frac{K_A}{K_B U_{rw}} \dot{\alpha}_{s_d} + \alpha_{s_d} - \frac{K_d}{U_{rw}} \dot{\tilde{\alpha}}_s - K_p \tilde{\alpha}_s - U_{rw} K_i \int \tilde{\alpha}_s dt \quad (6.24)$$

Notice that this controller includes division by the velocity, which can give infinite or significant high values as the velocity goes to zero. To prevent this effect the velocity used in the controller need to be lower bounded $U_{rw} \geq U_{min} > 0$.

In addition, when the actual relative velocity is negative, the control structure is no longer valid. At the same time, the system becomes uncontrollable as the relative velocity goes to zero. Hence for velocities under a desired bound, the controller is reduced to a simple reference feed forward $\delta_t = \alpha_{s_d}$.

Relation between time domain and non-dimensional system

By introducing the controller in the original system, an expression for the natural frequency ω_n^* and damping ratio ζ^* of the close loop system can be found. It can be shown that the following relation holds (not considering the effect of the integrator):

$$\omega_n^* = U_{rw} \omega_n \quad (6.25)$$

$$\zeta^* = \zeta \quad (6.26)$$

The equations show that the natural frequency varies linearly with relative velocity, while the damping ratio ζ^* remains constant and equal to the non-dimensional system. The expression for the natural frequency ω_n^* is useful for reference model adjustment, such that feasible tracking is ensured. It is also applied to check that there is sufficient bandwidth separation between the inner tail angle loop and the outer sail AOA loop within the span of relative velocities. It is assumed that when the relative velocity exceeds the desired operation range, the tail angle is set to zero.

6.2.3 Heading controller

A heading controller is developed to control the heading and course of the sail-drone, using the same approaches as for the sail AOA controller. As the method is the same, see section 6.2.2 for more details. Heading control is used in order to be able to control the system at low velocity when the course estimate becomes unreliable. When the course angle is reliable, crab angle compensation can be used in order to create course control. Alternative controller approaches can be found in [7] where integrator backstepping is applied, while in reference [8] \mathcal{L}_1 adaptive control theory is used. The approach used here builds on the idea that the sail system and vessel are essentially similar.

Extracting the nonlinear heading dynamics from the simplified model in equation 4.61 give:

$$M_{66}\ddot{\psi} = (N_r - K_C u_{rc})\dot{\psi} - (X_{\dot{u}} - Y_{\dot{v}} + K_D)u_{rc}v_{rc} + K_D u_{rc}^2 \delta_r \quad (6.27a)$$

$$K_C = k_{CL_r}(r_{brAC,x}^b)^2 \quad (6.27b)$$

$$K_D = k_{CL_r} r_{brAC,x}^b \quad (6.27c)$$

Where $(X_{\dot{u}} - Y_{\dot{v}})u_{rc}v_{rc}$ is the destabilizing Munk moment [16]. From Figure 4.10a and the discussion in section 4.2.5 the saildrone is observed to be straight line stable due to the counteracting term given by the rudder. Assuming that the rudder reduce the effective Munk moment, such that it can be neglected and the sway velocity $v_{rc} \approx 0$:

$$M_{66}\ddot{\psi} = -(K_C u_{rc} - N_r)\dot{\psi} + K_D u_{rc}^2 \delta_r \quad (6.28)$$

Further, the linear damping term $N_r\dot{\psi}$ can be canceled using feedback linearization principles. Alternatively, the linear damping term can be viewed as a disturbance. Notice that the damping term is dissipative and will not destabilize the system. Using both of this alternatives result in the dynamic in equation 6.29 which can be recognized as a similar structure as the sail AOA dynamics.

$$M_{66}\ddot{\psi} = -K_C u_{rc}\dot{\psi} + K_D u_{rc}^2 \delta_r \quad (6.29)$$

Equation 6.29 can be transformed into a velocity invariant system using the same transform as described in section 6.2.2, with $L = 1$, $U = u_{rc}$ and tracking error $\tilde{\psi} = \psi - \psi_d$. Where the linear controller of the resulting velocity invariant linear system can be described as:

$$\delta_r = \frac{M_{66}}{K_D} \frac{d^2\psi_d}{d\Gamma^2} + \frac{K_C}{K_D} \frac{d\psi_d}{d\Gamma} + K_d \frac{d\tilde{\psi}}{d\Gamma} + K_p \tilde{\psi} + K_i \int \tilde{\psi} d\Gamma \quad (6.30)$$

The controller gains can be selected using the pole placement algorithms in equation 6.31-6.33 together with the natural frequency and damping ratio. Notice that K_D is negative as it includes the x-component of the vector from {b} to the rudder AC.

$$K_p = \frac{\omega_n^2 M_{66}}{-K_D} \quad (6.31)$$

$$K_d = \frac{2\zeta\omega_n M_{66} - K_C}{-K_D} \quad (6.32)$$

$$K_i = \frac{\omega_n}{10} K_p \quad (6.33)$$

Transforming the controller back to the time domain gives the gain-scheduled controller in equation 6.34, where the scheduling variable is the relative surge velocity with respect to the ocean current u_{rc} . An optional feedback linearization term is also included (last term) in the controller; however, the effect of the term was observed to be neglectable during testing.

$$\delta_r = \frac{M_{66}}{K_D u_{rc}^2} \ddot{\psi}_d + \frac{K_C}{K_D u_{rc}} \dot{\psi}_d + \frac{K_d}{u_{rc}} \dot{\tilde{\psi}} + K_p \tilde{\psi} + u_{rc} K_i \int \tilde{\psi} dt - \frac{N_r}{K_D u_{rc}^2} \dot{\psi} \quad (6.34)$$

Notice that the sign of δ_r is negative when $\psi_d > \psi$. Hence the sign needs to be changed to comply with the implemented rudder model. It requires a positive input to create a positive change of heading, as defined in section 4.1.7.

Similar to the tail, the rudder deflection angle is limited due to the physical limitation of the actuator. Hence an integral anti-windup scheme is required here as well.

The natural frequency ω_n^* and damping ratio ζ^* of the closed loop system have the same property as discussed at the end of section 6.2.2. This means that the bandwidth changes linearly with the relative surge velocity, and the following relations hold:

$$\omega_n^* = u_{rc} \omega_n \quad (6.35)$$

$$\zeta^* = \zeta \quad (6.36)$$

Smallest and largest signed angle

As opposed to common practice, the controller does not include the smallest signed angle (ssa) function. Which map an angle, here $\tilde{\psi}$, into the interval $[-180^\circ, 180^\circ]$ in order to avoid turns over 180° . This has two reasons: First, upwind jibing requires the use of the largest angle between the reference heading and the actual heading. Second, the ssa-function was observed to cause an initial turning rate and sometimes a complete turn in the opposite direction due to the low tracking error (as the error angle input to ssa remains small). Due to these reasons, the ssa-function was placed before the reference model, where it selects the reference command that will cause the shortest turn. When upwind jibing is required, the ssa-function is disabled, and the modulo operator is used to map the angle into the range $[-360^\circ, 360^\circ]$, which allows the longest angle to be used as reference input. The implementation was observed to cause no ill effects under simulation. The implementation is described in algorithm 1, where the path following algorithm determines if ssa should be disabled or not, as shown in figure 5.2. Notice that the bias introduced by using the heading measurement is removed again.

Algorithm 1 Smallest and largest signed angle

```

Input:  $\psi_{ref}$ ,  $\psi$ ,  $ssa_{on}$ 
Output:  $\psi_{ref}^*$ 
if  $ssa_{on} == \text{true}$  then
     $e_{ssa} \leftarrow ssa(\psi_{ref} - \psi) = mod(\psi_{ref} - \psi + \pi, 2\pi) - \pi$ 
     $\psi_{ref}^* \leftarrow e_{ssa} + \psi$ 
else
     $e_{lsa} \leftarrow mod(\psi_{ref} - \psi - 2\pi, 4\pi) - 2\pi$ 
     $\psi_{ref}^* \leftarrow e_{lsa} + \psi$ 
end if
```

6.3 Sail propulsion

The proposed method for offline optimization of the sail AOA is presented in this section, together with a method to estimate the sway force given by the sail.

In [3] the authors map out the optimal sail angles of a fully actuated sail, using measurement data of the forces acting on the sail. This idea is similar to what is proposed in this section, except here, the lift and drag models are exploited to give the optimal sail AOA.

6.3.1 Optimal sail AOA

The sail system is underactuated, where the tail allows the sail AOA to be controlled within a limited range. Hence, to maximize propulsion along the heading

or course direction, the optimal sail AOA need to be found for all operation conditions. To do this, an expression for the sail-induced forces acting along a reference frame's axis is needed. This can be achieved using the already established expressions for $\angle A$, $\angle B$ and F_{LD} in section 4.2.2. For convenience, the body frame is used as a reference here. Utilizing these expressions together with figure 6.1, the forces can be decomposed into the reference frame:

$$\begin{bmatrix} F_x \\ F_y \end{bmatrix} = F_{LD}(\alpha_s, U_{rw}) \begin{bmatrix} \cos(\varepsilon + \angle B) \\ \sin(\varepsilon + \angle B) \end{bmatrix} \quad (6.37)$$

(6.38)

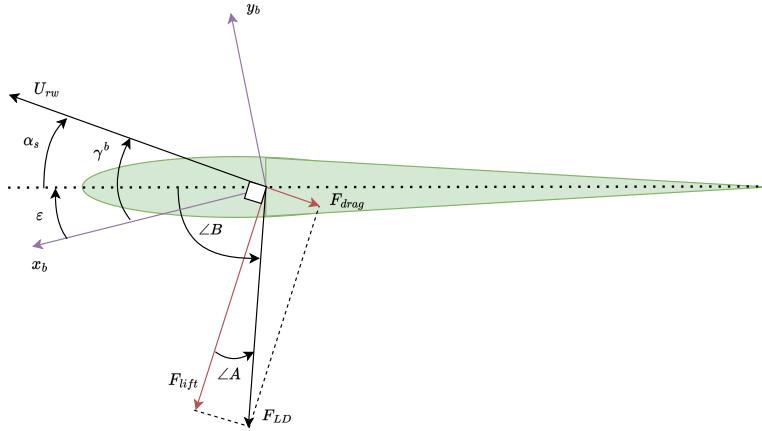


Figure 6.1: Sail induced forces

Further, the relations $\xi = \psi + \epsilon$ and $\beta_{rw} - \xi = \alpha_s$ can be introduced, and the decomposition angle can be rewritten as in equation 6.39. While the magnitude of the force vector is described by equation 6.40.

$$\varepsilon + \angle B = \underbrace{\beta_{rw} - \psi}_{\gamma^b} - \frac{\pi}{2} - \tan^{-1} \left(\frac{C_D(\alpha_s)}{C_L(\alpha_s)} \right) \quad (6.39)$$

$$F_{LD}(\alpha_s, U_{rw}) = \frac{1}{2} \rho U_{rw}^2 S \sqrt{C_D(\alpha_s)^2 + C_L(\alpha_s)^2} \quad (6.40)$$

The angle γ^b is the angle between the reference frame x-axis and the direction of the relative velocity in $\{n\}$, as seen in figure 6.1. Actually, this relation is generic for any reference frame, hence it will be valid for any $\gamma = \beta_{rw} - x$, where x is the angle of some vector direction in $\{n\}$.

Separating out the constants and the relative velocity, the forces acting along the reference frame x- and y-axis can be decomposed into two force coefficients:

$$\begin{bmatrix} F_x \\ F_y \end{bmatrix} = \frac{1}{2} \rho U_{rw}^2 S \begin{bmatrix} C_x(\gamma, \alpha_s) \\ C_y(\gamma, \alpha_s) \end{bmatrix} \quad (6.41)$$

The expression shows that the sail AOA that maximizes the force along the x-axis of the reference frame can be found by maximizing the force coefficient $C_x(\gamma, \alpha_s)$ for every $\gamma \in [-180^\circ, 180^\circ]$. Here α_s will have a limited range due to the dynamics constraining the angles of attack.

Equation 6.41 shows that the optimal sail AOA that maximizes the force along a specific reference vector only depends on γ . This implies that a solution for the optimal sail AOA is the same for any reference frame, as γ is generic. Hence an optimal solution can both be used to maximize the force along the heading and course direction.

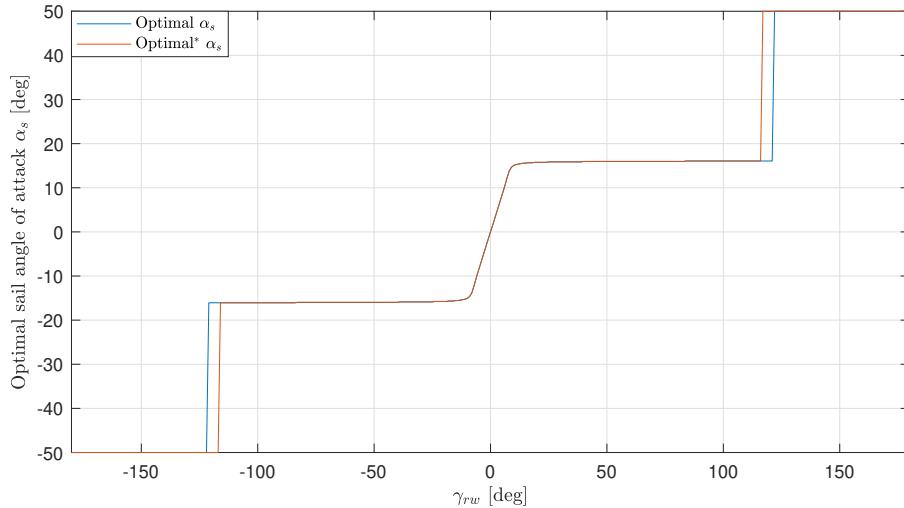


Figure 6.2: Optimal sail AOA

Optimizing the sail AOA within the range $\alpha_s \in [-50^\circ, 50^\circ]$ with a resolution of 1 degree for all $\gamma \in [-180^\circ, 180^\circ]$ give the result shown in figure 6.2. This solution is stored in a lookup table such that for any saildrone course/heading and relative velocity, the optimal sail AOA can be selected. The table will be discrete; however, the sail AOA reference model will provide the controller with a smooth reference to track.

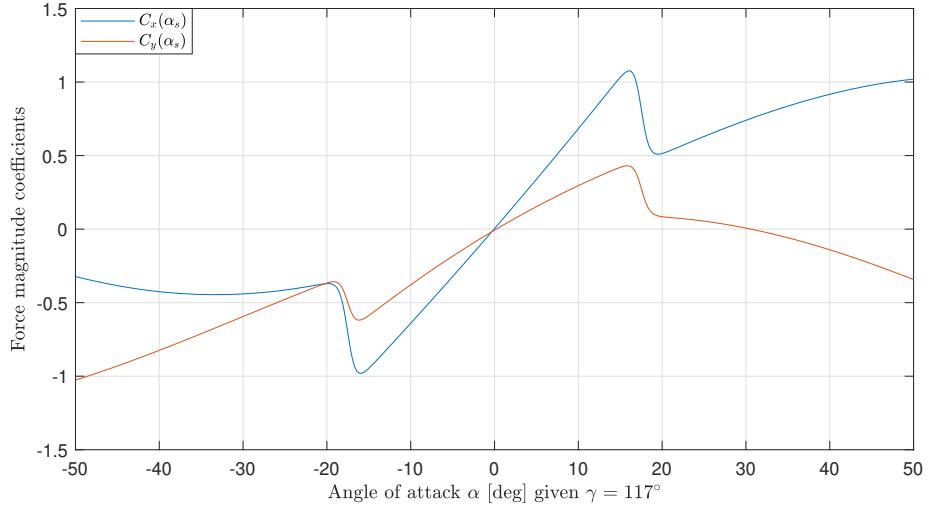


Figure 6.3: Force coefficients with respect to reference frame

Figure 6.2 reveal that there are 2 separated operation modes. The first mode maximizes forward propulsion using lift, while the second mode relies on both lift and drag. Further in this thesis, these are referred to as lift and drag mode, respectively. Close to the transition between these modes, there will be two distinct peaks in the total force coefficient plot as shown in figure 6.3. Figure 6.2 shows 2 optimal AOA for the sail, where the alternative "optimal*" α_s^* takes into account the two dominating peaks close to transition, together with the resulting force along the y-axis. When the difference is less than 10%, the option creating the least force along the y-axis is selected. This does not change much relative to "optimal α_s " which only take into account the forward force coefficient.

Figure 6.4 shows how the sail system changes between the optimal angles plotted in figure 6.2 for a circular path. Note that the optimal AOA was implemented with a threshold between lift and drag mode to avoid switching back and forth between the two.

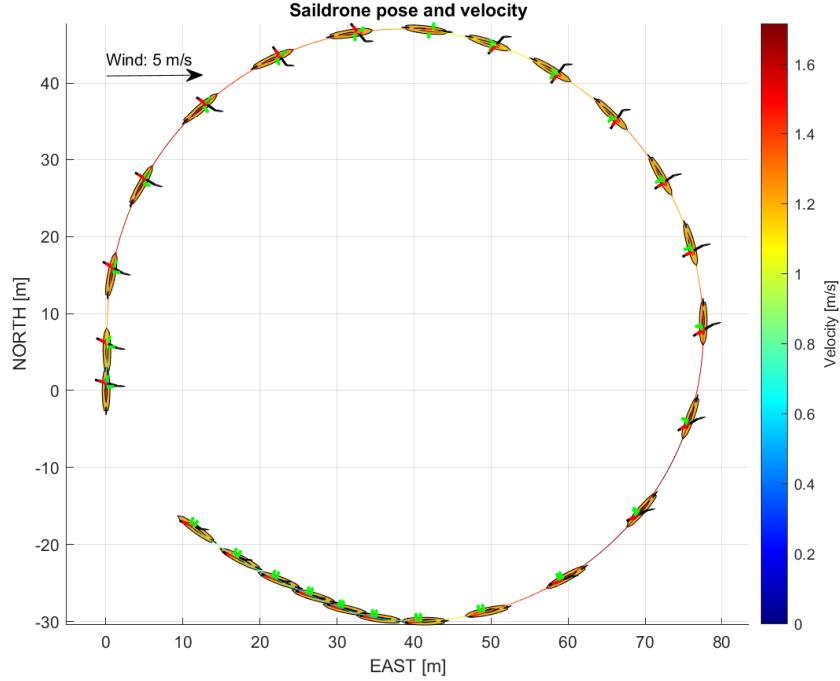


Figure 6.4: Sprial test for sail AOA

6.3.2 Accuracy and dominating dynamics at high sail AOA

The sail system heading dynamics used for controller design in section 6.2.2 is only valid for a limited range of sail AOAs. The assumption that $\alpha_s \approx \delta_t$ tends to hold also at higher AOAs as seen in figure 8.11, where the difference is 5.5 degrees when the $\alpha_s = 50$. However, when the sail AOA goes beyond stall, the moment contribution of the sail will be more evident. This means that the used model will capture less of the dominating dynamics. As a result of this, the controller was observed to oscillate for $\alpha_s > 65^\circ$ because the tail AOA needs to be close to or over stall to maintain the desired sail AOA. Despite this, the figure 8.11 and 6.9 shows a stable behavior at $\alpha_s = 50^\circ$ even when influenced by the system couplings and high roll rate. Due to this last observation, the drag mode was kept.

Notice that a physical system will be exposed to other disturbances and nonideal wind- and current- conditions. In addition, operating beyond stall can yield significant changes to the dynamics due to turbulent flow, as the flow is no longer attached to the foil. The behavior can be seen in the moment measurements in figure B.3, which has a large spread. The simulations show that the system remains stable also at higher sail AOA, but in reality, the turbulent flow might make this operation range uncontrollable. In this case, the lost drag mode operation range can be compensated by using only lift mode and a "wider" downwind crossing

region. The regions are further discussed in section 7.1.

6.3.3 Sail forces estimate

Besides the optimization, the expression in equation 6.41 can be used to estimate the sway force caused by the sail system. Where this estimate can be utilized to create a compensation term such that the force can be canceled using the keel AOA. This is possible as the force components along the reference frame axes are a function of U_{rw} , γ^b and α_s , which are available as measurements/estimates.

6.4 Course and heading autopilot

The sail propulsion produces forces along the desired course or heading direction, but sway forces are also introduced. However, the desired heading can be adjusted such that the keel AOA counteract these undesirable sway forces. This solution builds on the same idea which is used in reference [7]. However, in this reference, the ocean current is neglected and the sail systems are different, yielding a different compensation term.

Using the sway force estimate discussed in section 6.3.3 together with the assumption the sideslip angle β is approximate equal to the keel AOA α_k . The isolated sway force equilibrium is expressed by:

$$F_{y,keel}^b(U_{rc}, \alpha_k) = -F_{y,sail}^b(U_{rw}, \gamma, \alpha_s) \quad (6.42)$$

Where it is assuming the sideslip angle is small, meaning the keel AOA will remain in the linear lift region. This also implies that the lift of the keel is approximately parallel to the sway direction. Applying this simplification together with equation 6.41 and 4.53 give the following expression for the desired sideslip angle:

$$\beta_d \approx \alpha_{k_d} = \frac{U_{rw}^2 \rho_{air} S_s C_y(\alpha_s, \gamma)}{U_{rc}^2 \rho_{water} S_k C_{L_{\alpha,k}}} \quad (6.43)$$

Where S_k and S_s are the area of the keel and sail respectively. $C_{L_{\alpha,k}}$ is the linear lift coefficient of the keel. While the relative velocities are given at sail and keel AC.

Desired heading

A guidance system often gives the desired course of a craft as it is desirable to control the direction of travel with respect to $\{\mathbf{n}\}$. As a heading controller is utilized here, a relation between the heading and course is required. These are related by the crab angle β_{crab} as seen in figure 4.4 and expressed in the following equation:

$$\chi = \psi + \beta_{crab} \quad (6.44)$$

Hence the desired heading can simply be written as equation 6.45, where crab angle compensation is used as described in reference [16](page 363).

$$\psi_d = \chi_d - \beta_{crab} \quad (6.45)$$

To implement the desired sideslip angle, one can add and subtract the sideslip angle. Further, the sideslip error can be assumed to be small such that $\beta \approx \beta_d$. Then the heading controller input becomes:

$$\psi_d = \chi_d - \beta_{crab} + \beta - \beta_d \quad (6.46)$$

Which both take into account the desired course and sideslip. Notice that under the special case of no ocean current $\beta_{crab} = \beta$, the controller is reduced to:

$$\psi_d = \chi_d - \beta_d \quad (6.47)$$

Further, it is desirable to use sideslip angles within the linear lift coefficient region, which is the assumption used to deduce equation 6.43. Hence the desired sideslip is constrained within the linear lift region of the keel $\beta_d \in [-15^\circ, 15^\circ]$.

As opposed to a course controller, a heading controller can be used at any velocity, as the heading is always defined. Even close to zero, when the velocity and course measurements become too uncertain for control. However, at higher velocity, course control is desirable. This is achieved by applying a fading function that seamlessly changes the heading controller from course control to heading control. This is done by fading out the contribution from the crab- and sideslip- angles as the velocity decreases, because these terms are not defined or infinite at zero velocity. Hence at low velocity, the course reference from the guidance system is used directly as a heading reference for the controller. Notice that relative velocity is needed for the system to be controllable.

The final expression for the heading reference can be described by:

$$\psi_d = \chi_d + \sigma(u_{rc})(-\beta_{crab} + \beta - \beta_d) \quad (6.48a)$$

$$\sigma(u_{rc}) = \frac{1}{2} + \frac{1}{\pi} \operatorname{atan}(au_{rc} + b) \quad (6.48b)$$

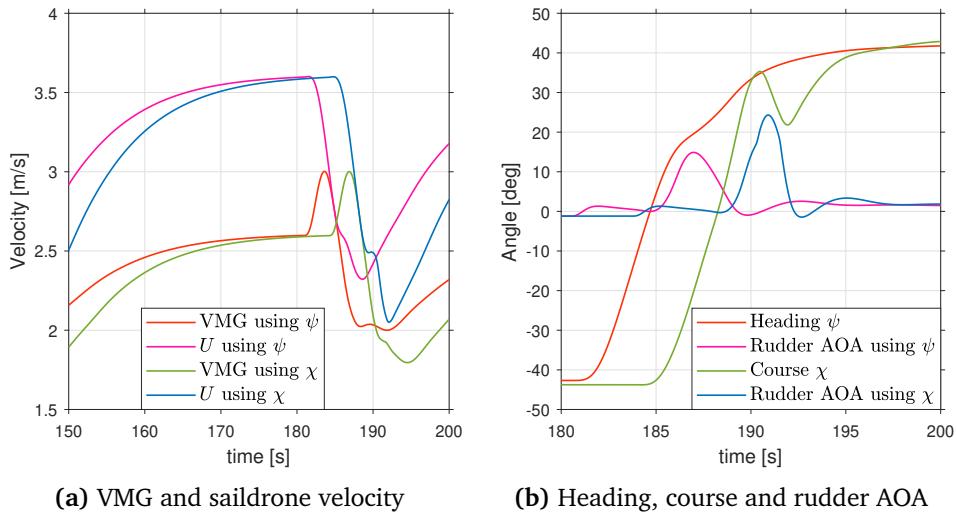
$$\beta_d \in [-15^\circ, 15^\circ] \quad (6.48c)$$

$$\text{if } u_{rc} < u_{rc,min} \Rightarrow \beta_{crab}, \beta, \beta_d = 0 \quad (6.48d)$$

Where $\sigma(u_{rc})$ is the fading function that exclude the sideslip and crab angle at low velocities. "a" is the rate of the transition phase, and "b" is used to adjust at which relative velocity the transition will take place. $u_{rc,min} > 0$ denote the minimum forward velocity at which the crab and sideslip angles are calculated. This should be at a lower velocity than when the transition will occur, but before any angle becomes infinite or not defined, preventing any infinite value from entering the heading controller after fading. The same idea of a smooth switching between heading and course control is used reference [39], where the fading is carried out by a sigmoid function.

6.5 Course or heading reference for optimal sail AOA

The lookup table for the optimal AOA, discussed in section 6.3.1, can be used to select a sail AOA that maximizes either the force along the course or the heading direction. In order to determine the best solution, the two alternatives were compared under normal operation. With a wind of 12 m/s at $\beta_w = 180^\circ$, for the courses $\chi = \{45, 90, 135\}$, with and without current of 0.5 m/s at $\beta_c = 90^\circ$. Here, the difference was observed to be neglectable. However, when crossing against the wind with a path angle of $\pi_p = 0$, it was observed that using the heading as a reference gave better results. This can be seen from the VMG and saildrone velocity U in figure 6.5a, where both values are considerably lower during the turn when using the course as a reference. Due to this result, the heading is used as a reference for the optimal sail AOA further in this thesis. Figure 6.5b shows the corresponding heading and course angles used as references, together with the rudder AOA for the two cases.



(a) VMG and saildrone velocity

(b) Heading, course and rudder AOA

Figure 6.5: System response during tacking with heading ψ and course χ as reference for optimal sail AOA

Notice that the heading and course are from 2 different simulations in figure 6.5b. Another argument for using the heading is that the course will generally point "outwards" during any turning maneuver, causing the sail system to provide an outwards sway force when maximizing propulsion along the course. This is undesirable as the outwards sway velocity is increased.

6.6 Controller tuning

6.6.1 Tail controller and reference model tuning

The tail angle controller gains were selected based on the controller developed through section 6.2.1. While a reference model was used to allow tracking control, reduce moment spikes and obtain a feasible moment command for the tail actuator. For the controller, a bandwidth of $\omega_{bw} = 38$ rad/s was selected together with a damping ratio of 1. Resulting in the following controller gains: $K_p = 488$, $K_d = 16.7$. While the feed forward gain was chosen to $k_{red} = 5/12$. The reference model natural frequency was set to $\omega_{nref} = 0.4\omega_n$ allowing some margin to ensure feasible tracking even with the simplifications. Also notice that the controller bandwidth has a good margin with respect to what a 100Hz controller potentially can track.

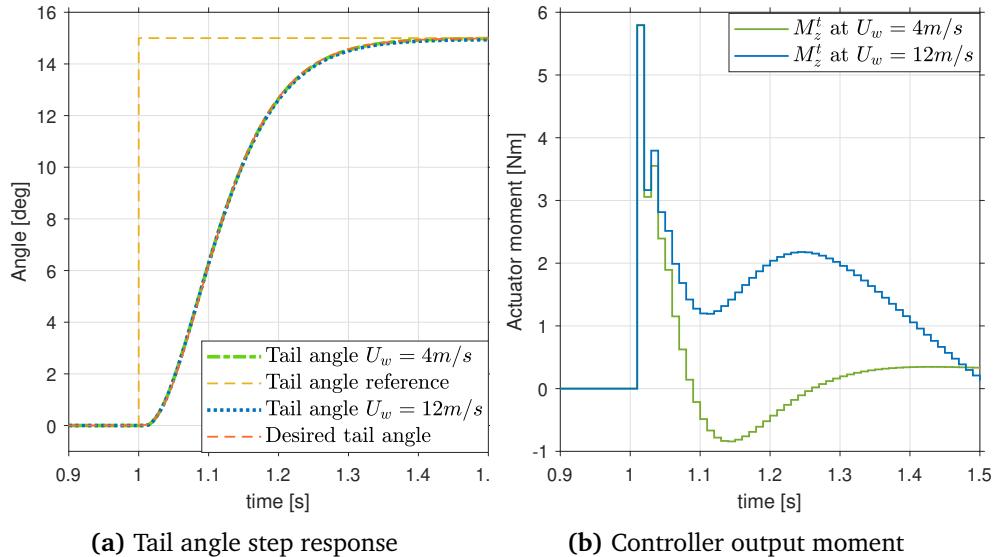


Figure 6.6: Tail angle step response at $U_w = \{4, 12\} \text{ m/s}$

Figure 6.6 show the response for a step of 15° at a wind velocity if $U_w = 4 \text{ m/s}$ and $U_w = 12 \text{ m/s}$ for the simulation model with all 6DOFs enabled. From figure 6.6a it is seen that the tracking is achieved under different wind velocities. Figure 6.6b show that the controller output, which is the desired actuator moment,

seems reasonable for a 100Hz controller. Compared to a setpoint controller, the output moment spike is significant reduced due to a smoother transition. In the implementation the actuator moment was limited to ± 50 Nm.

In figure 6.7 the system response for tacking under 15 m/s wind is shown. The optimal sail AOA is selected throughout the maneuver based on the material presented in section 6.3.1. Some oscillations are observed due to small moment adjustments at every discrete time step. Despite this, the controller output remains stable throughout the maneuver.

Notice that the sail AOA closely follows the tail angle as seen in figure 6.7a. This is an important property as it is one of the assumptions used under the derivation of the simplified sail system dynamics.

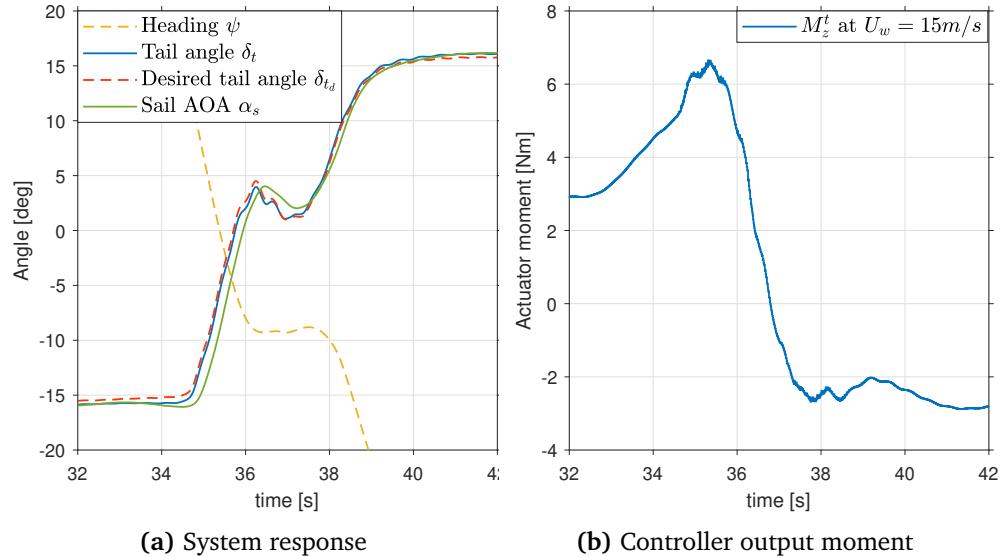


Figure 6.7: Angles and tail actuator moment under high wind tacking maneuver

6.6.2 Sail AOA controller and reference model tuning

The sail angle of attack controller gains was selected based controller developed in section 6.2.2. In addition a simple anti wind-up scheme was implemented, by stopping the integrator when the controller output exceeded the maximum tail angle. A bandwidth of $\omega_{bw} = 0.31$ rad/s and a damping ratio of 1 was selected for the velocity invariant closed loop system. Resulting in a closed loop natural frequency ranging up to $\omega_n^* = 7.8$ rad/s at a relative velocity $U_{rw} = 16$ m/s. Which also give sufficient bandwidth separation relative to the inner loop tail angle controller and its reference model. This is also satisfactory with respect to the controller frequency which was set to 40 Hz. Applying equation 6.21-6.23 together

with these parameters give the following controller gains: $K_p = 2.3$, $K_d = 11.8$, $K_i = 0.11$.

In addition, the reference model damping ratio was set to one to obtain a critical damped reference. While the natural frequency of the reference model was set to $\omega_{n_{ref}} = 0.6\omega_n^*$ in order to incorporate a good margin, ensuring feasible reference tracking.

The sail AOA step response for a step of 15 degrees (at time $t = 1$) under 4 and 12 m/s wind, is shown in figure 6.8. Where the sail system pivot point is fixed in the inertial frame both for the simplified and the simulation model. At the same time, the step is the input of the reference model. The response reflects the bandwidth change, allowing faster response when the relative velocity increases, where the repose at 12 m/s is significantly faster than at 4 m/s. The figure also indicates that the control law works well across the range of velocities, "stretching" the response as relative velocity decreases.

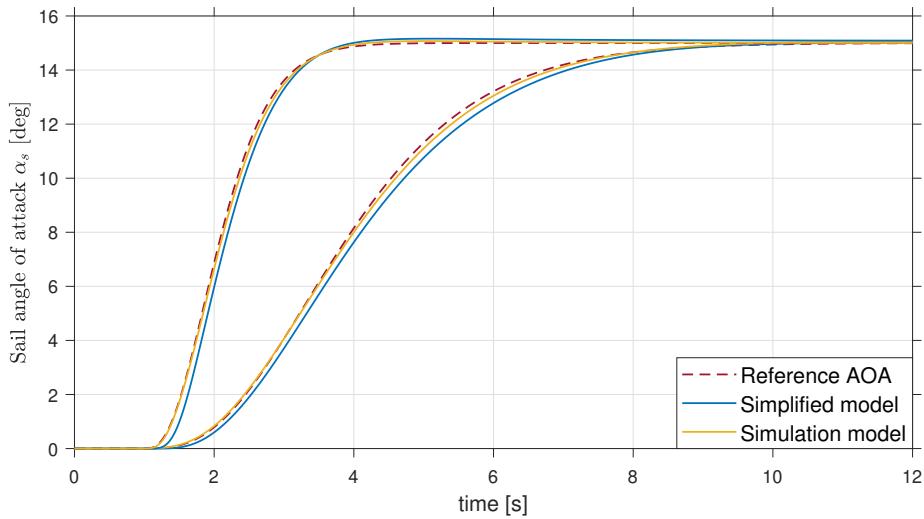


Figure 6.8: Sail AOA 15° step response at $U_{rw} = \{4, 12\}$ m/s

The system response of the simulation model under high wind conditions can be seen in figure 6.9. While the saildrone pose and velocity during the maneuver can be seen in figure 6.10. The roll angle peaks at 53 degrees and clearly affect the response, both directly and through the sail AOA reference model. This is mainly caused by the high roll rate, which affects the relative velocity at the sail AC. However, other effects also contribute, like the linear acceleration induced by the forces acting on the sail. Despite the disturbances, the controller has a stable behavior.

As the roll rate effect the reference model, an alternative is to use the relative

velocity with respect to wind at CO. This will reduce the transient roll rate interference on the reference model. However, this was not implemented due to only minor effects even under extreme conditions.

Figure 6.9 also show the stable behavior of the system when the sail AOA is set to 50 degrees. A more comprehensive discussion on high sail AOA can be found in sections 6.3.2. Also notice the stable behavior of the roll angle under these extreme conditions.

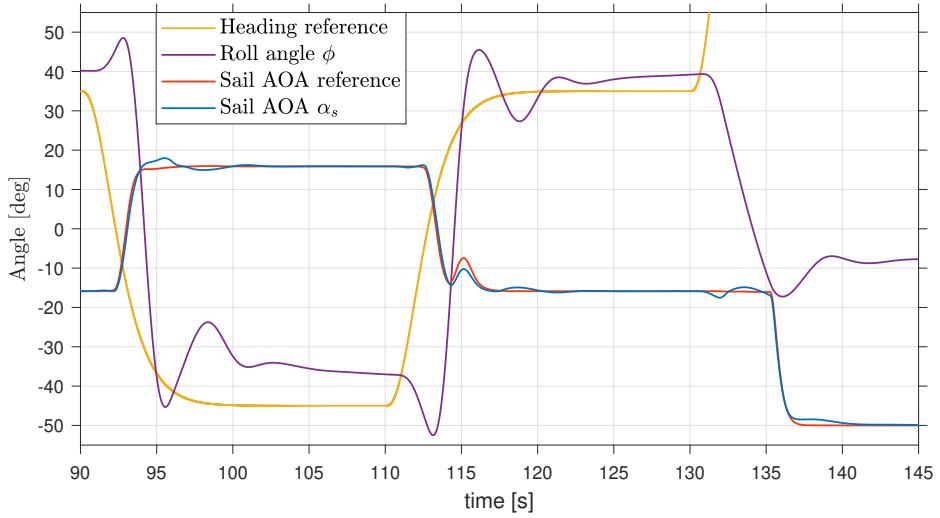


Figure 6.9: Sail AOA under heading change and wind speed $U_w = 16 \text{ m/s}$

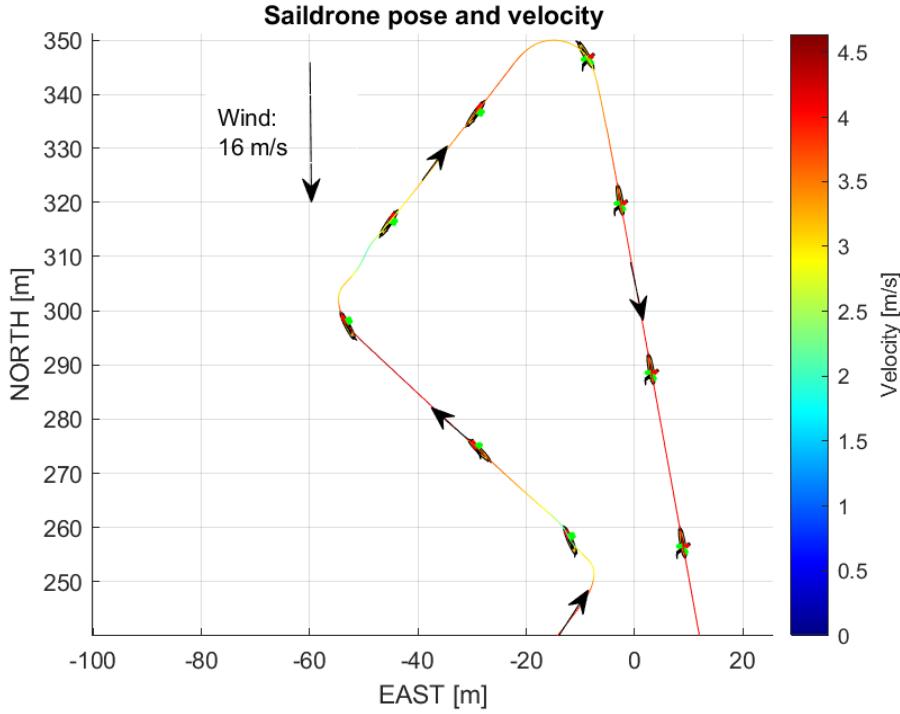


Figure 6.10: Saildrone pose and velocity under heading change and wind speed $U_w = 16 \text{ m/s}$

6.6.3 Heading controller and reference model tuning

The heading controller gains were selected based on the controller developed in section 6.2.3. In addition, a simple anti-windup scheme was implemented by stopping the integrator when the controller output exceeded the maximum rudder angle. A damping ratio of 1.5 and a natural frequency of $\omega_n = 0.67 \text{ rad/s}$ was selected, implying a bandwidth of $\omega_{bw} = 0.25 \text{ rad/s}$. Originally a damping ratio of 1 was selected; however, at maximum surge velocity, the controller was observed to create small undesirable oscillations after tacking maneuvers, caused by low bandwidth separation in combination with high roll rate during the maneuver. The natural frequency was hence reduced, but it was observed to compromise the tracking. Due to this combination, the damping was instead increased to 1.5 as an alternative way to reduce the bandwidth. This results in a slower overdamped system; meanwhile, the extra damping increase robustness, which is desirable when considering all disturbance terms excluded when developing the controller. At the same time, the slower response is permitted as the most effective turns are those where the rudder AOA is kept below the stall angle.

Using equation 6.31-6.33 the following controller gains were obtained: $K_p = 1.5$, $K_d = 3.4$, $K_i = 0.1$. Where the bandwidth ranges up to $\omega_{bw}^* = 1.63 \text{ rad/s}$ at a

relative surge velocity of $u_{rc} = 6.5$ m/s. Which is sufficient with respect to the rudder time constant of $T_r = 0.5$ s and controller frequency set to 40 Hz.

The heading reference model damping ratio was set to one to obtain a critical damped reference. At the same time, the natural frequency of the reference model was set to $\omega_{n_{ref}} = 0.5\omega_n^*$ in order to incorporate a good margin during lower surge velocity. At low velocity one want to utilize the full potential of the heading dynamics as the dynamics is slow. While at a higher velocity, one can make sharper turns due to the increased bandwidth, but smooth turns are preferred to avoid stalling the rudder. Hence at higher velocity the reference model natural frequency was constrained to $\omega_{n_{ref}} = \frac{2\pi}{10}$. Which can be summarized as follows:

$$\omega_{n_{ref}} = \min(0.5\omega_n^*, \frac{2\pi}{10}) \quad (6.49)$$

Figure 6.11 and 6.12 show the closed-loop step response for a heading change of 45 degrees under 4 and 12 m/s side wind. The average surge velocity under these maneuvers is 1.1 and 4.8 m/s, respectively. Here the simulation model is used with all DOFs enabled. As the wind angle is $\beta_w = 90^\circ$ and the heading change only with 45° from north, the sail remain in lift mode and the roll rate remains low as it is mainly effected by the rudder deflection. This is a best-case scenario, where the controller tracks the desired heading with minimal error. The smooth rudder angle shown in the figures also indicates stable controller behavior. Notice that rudder AOA remains small and that a smaller deflection is needed at higher surge velocity as the induced forces are higher.

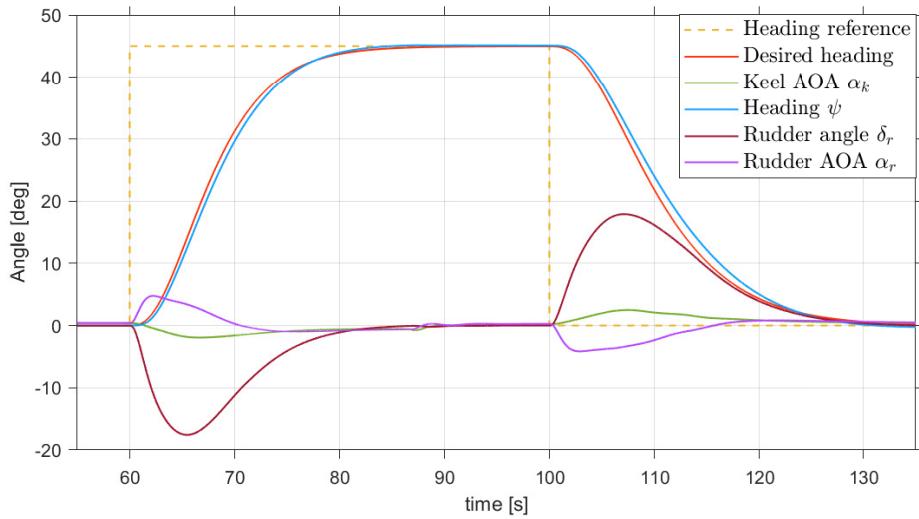


Figure 6.11: Heading dynamics step response under 4 m/s wind at $\beta_w = 90^\circ$

Similar to the sail AOA closed loop dynamics one can observed the "stretched"

response given by the controller and the reference model by comparing figure 6.11 and 6.12.

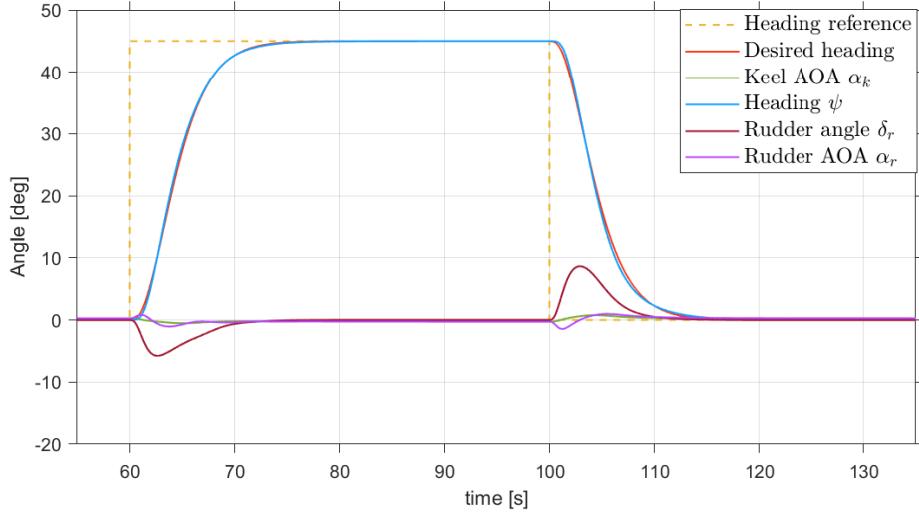


Figure 6.12: Heading dynamics step response under 12 m/s wind at $\beta_w = 90^\circ$

Another important observation from figure 6.11 and 6.12 is that the keel AOA remains small. This also implies that the sideslip angle remains small, which is one of the underlying assumptions used when deriving the simplified model.

In figure 6.13 the heading dynamics during upwind tacking under extreme wind condition is shown. During the tacking maneuver, the wind direction relative to the saildrone changes, such that the sway forces given by the sail switch over to the opposite side. This effect, together with the keel forces and rudder deflection, causes a high roll rate that effectively stalls the rudder. From the figure, it can be seen that the stalled rudder causes an undesirable effect where the heading changes in the opposite direction. Despite the effect, the controller is observed to have stable behavior, and the heading converges back to the desired heading. A solution for this problem can be to decrease the reference model turning rate, resulting in a slower transition and a lower roll rate. However, this will lead to a longer time spent facing the wind with no propulsion other than kinetic energy. It means that there will be a trade-off here. This effect is not studied further, but performance can be gained by reducing this effect. For example, by applying a model predictive controller which can take the role dynamics into account. Notice that the keel AOA remains small also in this case.

Another effect can be seen in figure 6.14, where an upwind jibing maneuver is executed. Here it is clear that the large reference change makes tracking infeasible. However, this can be solved by implementing a rate constraint in the heading reference model. The constraints will need to vary with velocity to give the best

result. This was not implemented as the controller was observed to converge to the desired heading with a stable response.

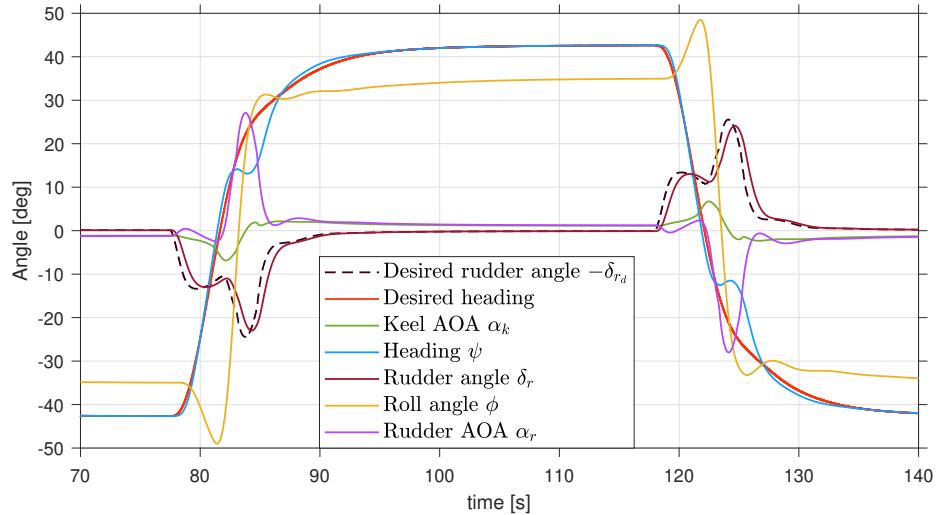


Figure 6.13: Heading dynamics during upwind tacking under 15 m/s wind

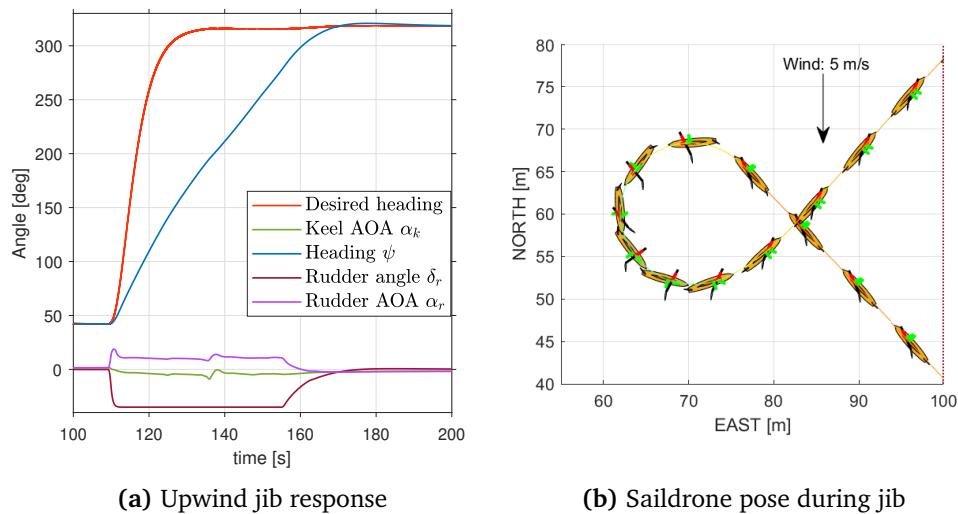


Figure 6.14: Upwind jib at 5 m/s wind at $\beta_w = 180^\circ$

6.6.4 Tuning summary

The tuning parameters for the controllers and their reference models are summarized in table 6.1.

Table 6.1: Controller and reference model parameters

Controller	Description	Symbol	Value	Unit
-	Tail angle controller frequency		100	Hz
	Sail AOA controller frequency		40	Hz
	Heading controller frequency		40	Hz
Tail angle	Proportional gain	K_{p_t}	488	
	Derivative gain	K_{d_t}	16.7	
	Reference model natural frequency	$\omega_{n_{ref,t}}$	23.4	rad/s
	Reference model damping ratio	$\zeta_{ref,t}$	1	
Sail AOA	Proportional gain	K_{p_s}	2.3	
	Derivative gain	K_{d_s}	11.8	
	Integral gain	K_{i_s}	0.11	
	Reference model natural frequency	$\omega_{n_{ref,s}}$	varying	rad/s
	Reference model damping ratio	$\zeta_{ref,s}$	1	
Heading autopilot	Proportional gain	K_{p_h}	1.5	
	Derivative gain	K_{d_h}	3.4	
	Integral gain	K_{i_h}	0.1	
	Reference model natural frequency	$\omega_{n_{ref,h}}$	varying	rad/s
	Reference model damping ratio	$\zeta_{ref,h}$	1	

Guidance system

In this chapter, the guidance system is presented, which consists of 3 modules: Path following, collision avoidance and path management. Here the path following algorithm deal with the motion control problem and ensure that every position in $\{n\}$ is reachable by selecting the desired course and how to progress along the local path. The collision avoidance algorithm ensures that obstacles are avoided if they block the path. While the path manager updates the waypoints as the sail-drone progresses.

In addition, the method utilized to make any direction of travel possible through crossing is presented, which is the foundation for solving the motion control problem and making any position in $\{n\}$ reachable. This chapter also describes the novel method developed to incorporate ocean current into the calculation of the optimal crossing courses. Such that the optimal course exploits both the wind and ocean current. This work also shows that the ocean current effect both the obtainable velocities and the regions where crossing is required.

7.1 Feasible velocities

For sailboats, a velocity space polar diagram is often used to represent the feasible velocities. The diagram shows the feasible steady-state velocities along the different constant course angles relative to the wind direction, like the one in figure 7.1. Different methods for estimating the diagram exist: real-time estimation using data capture in reference [40] and polar diagram approximation using interval algorithm [41], to mention a few. The diagram of the feasible velocities will be referred to as FVD (Feasible velocity diagram) in this thesis.

In this thesis, a simulation-based equilibrium/steady-state analysis is used to make an offline estimate of the diagram. Where a finite set of simulations along the different course angles are executed until the simulation model presented in section

4.1 reaches an equilibrium point. Where the sail propulsion equals the dissipative forces, resulting in constant velocity. To limit the number of simulations, the symmetry of the vessel is utilized, implying that the FVD will be symmetric. In addition, a simulation horizon of 200 seconds and a course angle resolution of 1 degree is used for each wind velocity $U_w \in [4, 8, 12, 16]$ m/s. The result can be seen in figure 7.1 where the sail AOA optimizer is constrained to use $\alpha_s \in [-50^\circ, 50^\circ]$. The entire equilibrium analysis is carried out with zero ocean current.

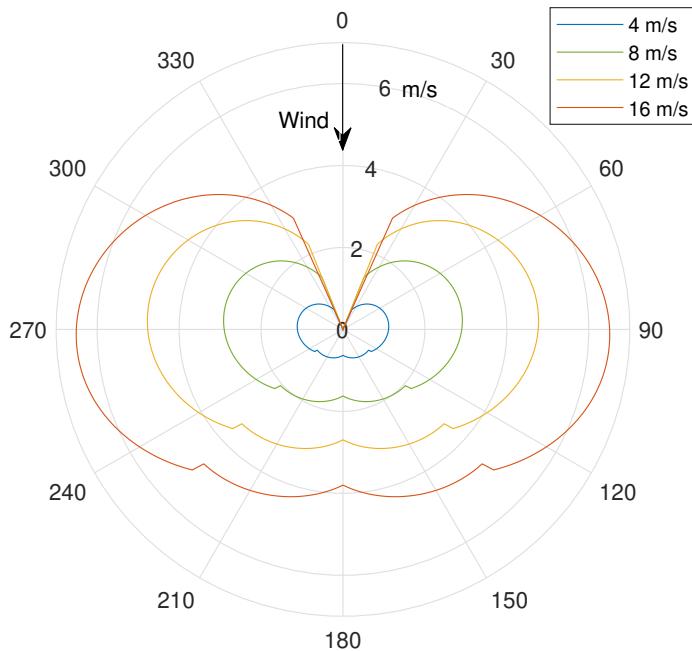


Figure 7.1: Steady state analysis for varying wind speed

The two steps in the velocity plots at approximately $\pm 135^\circ$, are caused by the change of operation mode, transitioning between drag and lift mode. Seen from figure 6.2 the optimal sail AOA transition phase is around $\pm 120^\circ$. However, the relative velocity v_{rw} changes as the velocity of the saildrone increases hence the different location in the FVD. The sudden drop indicates that the lift mode operation range could be slightly larger. The drop is caused by the reduced sway force, which causes a reduced keel AOA, meaning the forward component of the keel lift is reduced. Notice that increased keel AOA also increases drag, and the keel contribution is not entirely positive.

The diagram also shows that the drag mode region $\in [135^\circ, 225^\circ]$ does not provide a significant change to the maximum velocity downwind. However, it allows higher velocity with fewer jibing maneuvers as the downwind optimal angle is smaller (the angle between wind direction and the course direction that provides the

highest velocity along the wind direction).

Dead zone

Figure 7.1 also shows the famous dead zone, which is the range of directions that do not provide forward velocity when sailing upwind [42]. This indicates that alternating between two feasible courses is required to travel in these directions. The dead zone is also referred to as the no-go or no-sail zone in literature. The dead zone can also be observed from the optimal angle; when going too steep against the wind, the optimal sail AOA will go toward zero to minimize the force pushing the saildrone backward.

From figure 7.1 one can also observe that the maximum downwind velocity is actually not directly along the wind direction (called dead run). This makes it preferable to jib when going downwind. From a sail AOA perspective, this is also positive as a direct downwind direction with small perturbation will cause the optimal sail AOA to alter between positive and negative angles.

7.1.1 Convexification of velocity space using tacking and jibing

In order to make all positions in $\{n\}$ reachable, the saildrone has to be able to have an effective velocity in any direction. As seen from the non-convex FVD in figure 7.1 and the illustration in figure 7.2, a direct course towards the target is not possible within the dead zone. In addition, when sailing downwind, the highest velocity is not directly along with the wind in the dead run zone. Hence tacking and jibing are required to maximize the effective velocity towards the target position and make all positions reachable.

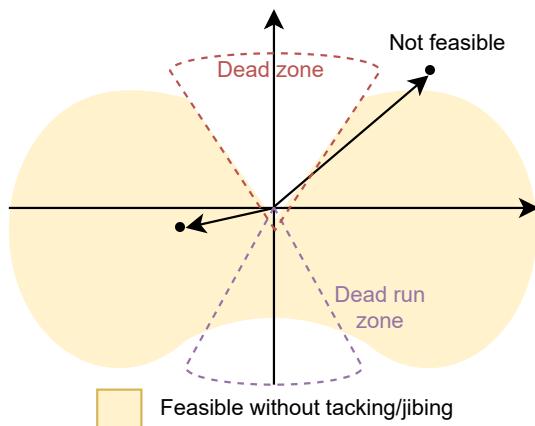


Figure 7.2: Generic feasible velocity space diagram (FVD)

Applying jibing or tacking enables the combination of the two legs of travel, each with an individual velocity, which combined gives a new effective velocity. This sailing principle will be applied here to make a convex version of the FVD. The convex version is denoted CVD (convex velocity diagram). The following discussion on convexification is adapted from the provided note on convexification by tacking mentioned in the preface.

Each leg of travel will represent a feasible velocity vector within the FVD, meaning $v_1, v_2 \in \text{FVD}$. Denoting the time spent in each leg as t_1 and t_2 . The distanced covered satisfy:

$$\bar{v}(t_1 + t_2) = t_1 v_1 + t_2 v_2 \quad (7.1)$$

Where the accelerations due to the tacking or jibing maneuvers are neglected.

Dividing by the period time $t_1 + t_2$ and defining the new variable λ :

$$\bar{v} = \frac{t_1}{t_1 + t_2} v_1 + \frac{t_2}{t_1 + t_2} v_2 \quad (7.2)$$

$$\lambda = \frac{t_1}{t_1 + t_2} \quad (7.3)$$

Notice that $\lambda \in [0, 1]$ for all positive values of t_1 and t_2 . Introducing the new variable into equation 7.1 give the following linear relation:

$$\bar{v} = \lambda v_1 + (1 - \lambda) v_2 \quad (7.4)$$

This implies that all effective velocities, which are linear combinations of the two vectors, can be obtained by varying the time spent in each leg. Hence all effective velocities on the line joining the tips of the velocity vectors are feasible, as illustrated in figure 7.3.

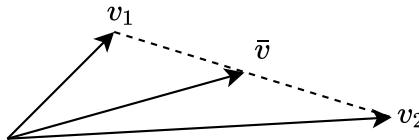


Figure 7.3: Linear combination of velocity vectors

Geometrically this means that any two points in or on the border of the FVD, can be joined by a line. By evaluating all linear combinations of point pairs within a two-dimensional set, it can be shown that the resulting shape (set) will be convex. This is exploited to define the CVD as the convex set of all velocities in the FVD. Formally this can be written as equation 7.5 and illustrated as in figure 7.4.

$$CVD = \{\lambda v_1 + (1 - \lambda)v_2 | \lambda \in [0, 1], v_1, v_2 \in FVD\} = \text{convex}(FVD) \quad (7.5)$$

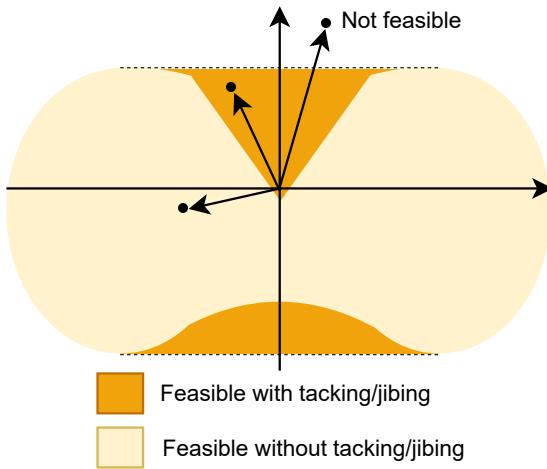


Figure 7.4: Generic convex velocity space diagram (CVD)

Further in this thesis, the effective velocity will be obtained by using positional constraints rather than time in order to limit the path width when crossing. In addition, this provides a simple measure of when to tack/jib without being affected by acceleration and other disturbances. The directions of the velocity vectors will be the same, but tacking/jibing is executed based on distance from the main path.

7.1.2 Optimal up- and downwind course

FVD and CVD provide the optimal course angles (relative to the wind direction) to maximize the effective velocity when sailing upwind or downwind under zero ocean current. These are simply the angles of the velocity vectors that span the border of the CVD in the respective direction. Which are the vectors from the origin to the end of the dotted lines in figure 7.4. Hence going up or downwind, the course will switch between these two angles to provide the highest velocity. While the resulting effective velocity vector will determine the composition of the two vectors.

Table 7.1 show the upwind and downwind optimal course angles. These are found by evaluating the data obtained from the equilibrium analysis of the specific sail-drone, which forms the FVD estimate in figure 7.1. The values are close to constant and independent of the wind velocity magnitude.

Table 7.1: Optimal course angle with respect to wind direction at zero current

Wind velocity [m/s]	4	8	12	16
Optimal course [deg] upwind	±44	±43	±43	±44
Optimal course [deg] downwind	±163	±163	±164	±164

7.1.3 Ocean current and feasible velocities

As established previously in this section, the feasible velocities (FVD) and the resulting CVD are based on the equilibrium point where the dissipative forces cancel the propulsion forces when the ocean current is zero. Ocean current is usually ignored when selecting the optimal course angle for crossing, judged by all the literature studied. However, the ocean current does affect the feasible velocities in $\{n\}$ due to the dissipative forces induced by the saildrone's hull, rudder and keel. Which by foil theory are given by the relative velocity between the saildrone and the ocean current. In the following section, the effect of ocean current will be introduced and discussed, while the performance gain is described in section 8.4. To the author's knowledge, this approach has never been investigated before. By introducing current into the CVD, one is able to select crossing courses that provide the fastest courses utilizing both the influence of wind and ocean current. In addition, the course angles are more likely to be feasible, which increases robustness.

Introducing ocean current

From figure 7.1 the velocity magnitude can be seen to scale with the magnitude of the wind $\|v_w\|$, but also as a function of the course angle relative the wind direction. Now, assuming the shape of the FVD, when the wind velocity is normalized, is given by a polar function of the course angle relative to wind $r(\chi - \tilde{\beta}_w)$. Which is illustrated in figure 7.5 and where $\tilde{\beta}_w = ssa(\beta_w + \pi)$ is the angle of the of a vector going against the wind. Further, the wind velocity magnitude $\|v_w\|$ inflate or scale the magnitude of the feasible velocities. This suggests that the magnitude of the feasible velocities is given by equation 7.6, while the resulting vectors are given by equation 7.7.

$$U_{FVD} = \|v_w\| r(\chi - \tilde{\beta}_w) \quad (7.6)$$

$$v^{FVD} = \begin{bmatrix} \cos(\chi_r - \tilde{\beta}_w) \\ \sin(\chi_r - \tilde{\beta}_w) \end{bmatrix} U_{FVD} \quad (7.7)$$

Equation 7.6 seems reasonable when considering figure 7.1, as the shape is somewhat preserved independent of the wind velocity. If equation 7.6 holds, the relation is homogeneous, and it can describe all FVD for the particular saildrone. For the coming derivation, it is assumed that this property holds. Or at least it

holds as a linear approximation between the measurement data presented at different wind velocities in figure 7.1, as the shape is slightly distorted at higher wind velocity.

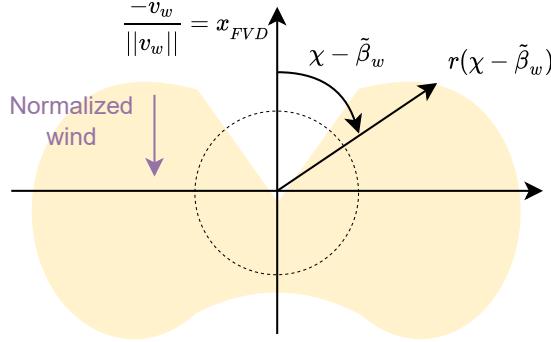


Figure 7.5: A normalized homogeneous FVD with $v_c = 0$

Considering the following 3 velocity vectors with respect to $\{n\}$ shown in figure 7.6: the velocity vector of the saildrone v , wind v_w and ocean current v_c . Here the superscript is dropped to keep the expression clean. As the FVD is given at zero ocean current, the vectors are transformed into a frame where the ocean velocity is zero, by subtracting the v_c from all vectors. This results in the new vectors:

$$v'_c = v_c - v_c = 0 \quad (7.8)$$

$$v'_w = v_w - v_c \quad (7.9)$$

$$v' = v - v_c = v_{rc} \quad (7.10)$$

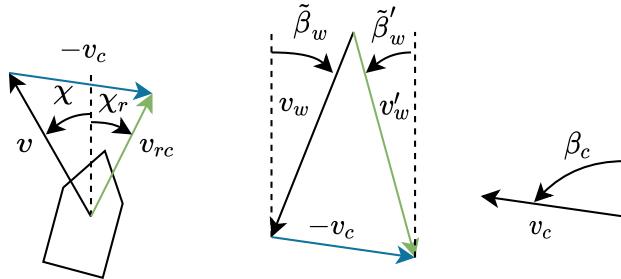


Figure 7.6: Velocity vector transformation

The resulting vector are shown in figure 7.6. Also notice that the angle of the relative velocity vector v_{rc} is the relative course χ_r of the saildrone with respect to the ocean current.

In the new perspective, the effective wind velocity is given by v'_w , which is the velocity of the wind with respect to the ocean current. The magnitude of this vector will give the inflation of the FVD and the angle of the vector will give the rotation to the FVD frame. This is the frame used for all FVD and CVD illustrations, where the direction of the effective wind is always pointing along the negative x-axis (following NED convention). Defining the rotation matrix from $\{n\}$ to FVD, which is an elementary rotation around the z-axis:

$$R_n^{FVD}(\tilde{\beta}'_w) = R_{FVD}^n(\tilde{\beta}'_w)^T = R_{FVD}^n(-\tilde{\beta}'_w) \quad (7.11)$$

Rotating the vectors into the FVD frame gives the following result:

$$v''_w = R_n^{FVD}(\tilde{\beta}'_w)v'_w \quad (7.12)$$

$$v''_{rc} = R_n^{FVD}(\tilde{\beta}'_w)v_{rc} \quad (7.13)$$

Within the new frame, the velocity magnitude is given by the relation established in equation 7.6 and illustrated in figure 7.5. Composing the magnitude with the unit vector of v''_{rc} give the feasible velocity vector $v^{*''}$:

$$v^{*''} = \frac{v''_{rc}}{\|v''_{rc}\|} \|v''_w\| r(\angle v''_{rc}) \quad (7.14)$$

Where $\angle v''_{rc}$ is the angle of the vector.

Transforming equation 7.14 back by applying the inverse operations yield:

$$v^{*'} = R_n^{FVD}(\tilde{\beta}'_w)^T v^{*''} \quad (7.15)$$

$$v^* = v_c + R_n^{FVD}(\tilde{\beta}'_w)^T v^{*''} \quad (7.16)$$

Introducing equation 7.14 into 7.16 and using the relation $\angle v''_{rc} = \chi_r - \tilde{\beta}'_w$, the following equation can be deduced:

$$v^* = v_c + R_n^{FVD}(\tilde{\beta}'_w)^T \frac{v''_{rc}}{\|v''_{rc}\|} \|v''_w\| r(\chi_r - \tilde{\beta}'_w) \quad (7.17)$$

Using the fact that the rotation matrix preserves the magnitude of the vectors and inserting equation 7.13:

$$v^* = v_c + \frac{v_{rc}}{\|v_{rc}\|} \|v_w - v_c\| r(\chi_r - \tilde{\beta}'_w) \quad (7.18)$$

Equation 7.18 reveals the effects of wind and ocean current. If $v_c = 0$, implying $\tilde{\beta}'_w = \tilde{\beta}_w$, it is observed that the wind contribute with 2 effects. First the magnitude $\|v_w\|$ scale the FVD, second the direction of the wind rotates the FVD in $\{n\}$. If $v_c \neq 0$, then it is observed that the ocean current shift the FVD by the v_c . In addition, it contributes in scaling and rotation of the FVD in $\{n\}$. This means that the

introduction of current; rotate and translate the region where crossing is required.

Another interesting perspective can be observed in the FVD frame. Transforming equation 7.18 into the FVD frame and using the relation in equation 7.19 give equation 7.20.

$$\frac{v_{rc}}{\|v_{rc}\|} = \begin{bmatrix} \cos(\chi_r) \\ \sin(\chi_r) \end{bmatrix} \quad (7.19)$$

$$\underbrace{R_n^{FVD}(\tilde{\beta}'_w)v^*}_{\check{v}} = \underbrace{R_n^{FVD}(\tilde{\beta}'_w)v_c}_{\check{v}_c} + \underbrace{\begin{bmatrix} \cos(\chi_r - \tilde{\beta}'_w) \\ \sin(\chi_r - \tilde{\beta}'_w) \end{bmatrix} r(\chi_r - \tilde{\beta}'_w) \|v_w - v_c\|}_{\check{v}_{rc}} \quad (7.20)$$

Equation 7.20 is illustrated in figure 7.7. Notice that \check{v}_{rc} is the vector that from the FVD/CVD origin.

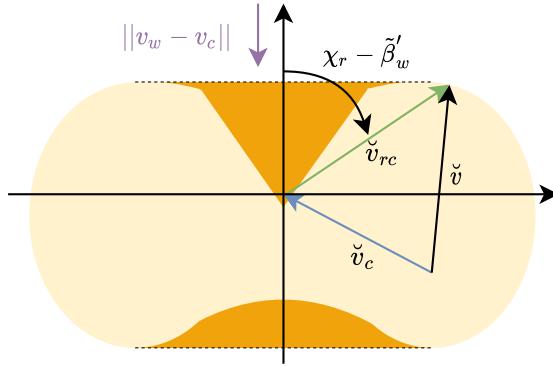


Figure 7.7: CVD including ocean current

Figure 7.20 show that any effective velocity \check{v} that require $\check{v}_{rc} \in (CVD-FVD)$ will have to be obtained by crossing. Where the optimal crossing courses are given by table 7.1, which shows that the courses are the same regardless of the effective wind velocity. Notice that these angles now representing \check{v}_{rc} . Defining the velocity vectors of the optimal crossing angles as $\check{v}_{rc,xR}$ and $\check{v}_{rc,xL}$, the resulting velocity vectors in $\{n\}$ are obtained by equation 7.21. While the crossing courses (χ_{x_R} and in χ_{x_L}) $\{n\}$ will be the angles of these vectors. With R and L referring to left or right of the resulting course.

$$v_x = v_c + R_n^{FVD}(\tilde{\beta}'_w)^T \check{v}_{rc,x} \quad (7.21)$$

This makes it rather simple to find v_x when v_c and $\tilde{\beta}'_w$ are estimated, as $\check{v}_{rc,x}$ only switch between two values dependent on the direction of travel relative to the wind.

On the other hand, determining whether crossing is needed is more involved. This is because both \check{v} and \check{v}_{rc} are unknowns and not free to select due to the sail propulsion. However, the desired course angle is known, which gives the direction of \check{v} . Assuming it is of interest to always maximize the velocity and that a feasible solution exists. Then a solution is found where the line that coincides with \check{v} intersects the border of the CVD. Hence \check{v}_{rc} can be found. Actually, as the border of the regions where crossing is needed are line segments, one only needs to look for a solution within these lines. Indicated by the dotted lines in figure 7.21. Because if $\check{v}_{rc,x}$ is not intersecting these line segments, crossing is unnecessary, and a direct course is the fastest.

7.2 Path following algorithm

The developed path following algorithm, which determines the desired course and how to best progress along the path given two waypoints, is presented in this section. The algorithm decides if tacking/jibing is required based on the current compensated CVD presented in section 7.1.3. When not required, the path is followed using the look-ahead-based LOS guidance law presented in section 2.6.1. When going against the wind, both tacking and jibing can be used. Here the algorithm selects tacking if the relative velocity with respect to the ocean current is over a selected limit. This is to ensure that the saildrone has sufficient kinetic energy to make the tack, preventing the system from becoming uncontrollable due to low velocity when exiting the tacking maneuver. This also applies to other maneuvers that require the saildrone bow to cross against the wind.

The presented algorithm has similarities to the algorithm found in reference [11], where course control is used when a direct course is not feasible due to the wind. However, the algorithm developed in this thesis differs by compensating for ocean current and allowing jibing (both up- and downwind). In addition, the algorithm in this thesis updated the previous waypoint with the current saildrone position when a direct course is feasible. Such that a new path segment is created, which can be solved in the exact same way as the previous segment. Furthermore, in reference [11] vector-field guidance is applied while a look-ahead-based LOS guidance is utilized here. The two guidance methods are essentially the same, where LOS guidance is a special case of vector-field guidance. For details see reference [16] page 353 to 361. The algorithm in [12] is also similar, it is roughly the same as [11], but it uses LOS as well.

An alternative approach to provide crossing capabilities using LOS is also presented. This approach is a simplification of the method used in reference [3]. Here the crossing is enabled by creating intermediate waypoints in a zigzag pattern. The approach is only for comparison and features only tacking and downwind maneuvers. Here a separate waypoint switching mechanism is used to change between

the local waypoints.

As the main method switches between course and LOS guidance, the guidance system features no integration; it is placed in the heading controller instead.

7.2.1 Course and LOS altering approach

As established in section 7.1.1 crossing is required to make any position in $\{n\}$ reachable. The proposed algorithm presented in the coming section switches between look-ahead LOS guidance and pure course control. When going upwind or downwind, using jibing or tacking, the algorithm operates in course control mode to ensure that the saildrone's course is kept optimal with respect to the wind and ocean current. When these maneuvers are not required, the algorithm uses LOS guidance to follow the direct path toward the target.

The working concept of the algorithm is presented here as an explanation, while the entire pseudo-code and overview of the states and parameters can be found in appendix A.

The algorithm is designed to run at a desired sampling rate, and does not require a specific state to be reached, e.g., position or constraint, to be reevaluated. Hence the optimal course given the current environmental- and saildrone-states is ensured as the saildrone progresses.

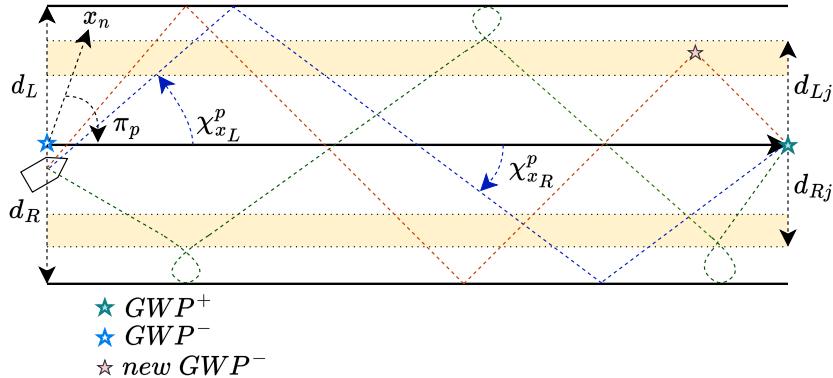


Figure 7.8: Crossing path

Figure 7.8 shows a crossing path together with some of the basic parameters used in the algorithm. The outer lines represent the path width constraints given by the distances d_L and d_R , where the tacking maneuvers are initiated. While the yellow regions represent the inner constraints where upwind jibing is initiated if required. These regions are adjusted using d_{Lj} and d_{Rj} . $\chi_{x_L}^p$ and $\chi_{x_R}^p$ are the crossing courses given in the path frame. The red and blue example paths util-

ize tacking, while the green upwind jibing path needs some extra space for jibing in order not to violate the outer constraint. $new\ GWP^-$ represent the update of GWP^- when a straight course to target is feasible without crossing. Notice that GWPs are used in the description here, but the algorithm is generic and can also receive WPs, as shown in figure 5.2.

Algorithm description

The working concept of the path following algorithm is presented here. The description follows the structure of the pseudo-code in algorithm 3. The algorithm basically consists of 4 main parts: Check if crossing is required and calculate the optimal crossing courses based on section 7.1, select the desired course reference, allow upwind jibing when required, and update algorithm states for the next iteration. Going into detail, the algorithm can be described as follows:

1. Compute the path angle π_p , direct course to target χ_t and the cross-track error from the path given by GWP^+ and GWP^- .
2. Use the path angle, ocean current and wind to determine if crossing is required based on the material established in section 7.1. If required, the optimal crossing angles are found using the same material. Notice that at this point, only the path direction is used, not the course directly to the target.
3. Rotate the course angles into the path frame such that one can distinguish between left and right of the path.
4. Select the desired course reference:
 - a. If the saildrone is crossing and was crossing in the previous iteration, then: Update the GWP^- with the current position if the direct course toward the target is feasible without crossing, by some margin. Such that the algorithm evaluates this new shorter path segment in the next iteration. Further, check if one of the path constraints is exceeded. In that case, switch to the other crossing course.
 - b. If transitioning from non-crossing to crossing: If the saildrone is within the inner path constraints for jibing, select the crossing course closest to the current course. Otherwise, the saildrone is further away from the path, and the course intersecting the path is selected.
 - c. If crossing is not required: LOS guidance is used if the saildrone is not executing an upwind jibing maneuver. If it does, then the direct course to target is used as the desired course.
5. Determine if upwind jibing is required: It is required when the new desired course originally requires the bow to cross against the wind direction (or v'_w which is more precise), while the velocity is not sufficient to make a tacking maneuver. A special case of this scenario is when the saildrone is crossing and was crossing in the previous iteration. Here, every course change originally requires the bow to cross the wind. Hence the velocity is only checked

when the saildrone is within the yellow regions shown in figure 7.8. Such that an upwind jib can only be executed when needed and without compromising the outer constraints.

6. Jibing specific operations:

- a. If upwind jibing is required: Find the offset angle $\in \{0, 2\pi\}$ that gives the longest turn when added to the desired course angle. Keep using the offset until the upwind jib is finished.
- b. Terminate the jibing maneuver: If the error between the desired course and the actual course is within a selected limit, the upwind jibing maneuver is terminated. Meaning the maneuver is finished, and the system starts using the smallest signed angle again.

7. Update the algorithm states for the next iteration.

In this algorithm, a conservative lower limit for the saildrone's relative velocity with respect to the ocean current is used to select between tacking or upwind jibing. This is a simple and conservative approach. However, a more advanced method is found in reference [32, 43], where the surge and heading dynamics are exploited to estimate the tacking maneuver exit velocity. Then the estimate is used to choose between two maneuvers.

A threshold was also implemented to avoid crossing with small traveling legs and fluctuation between a direct path and crossing. This can be done as there is a margin from the angle where crossing becomes desirable to where propulsion is lost or significantly reduced. However, a sufficient margin must remain to avoid issues when going directly to the target, as LOS can compromise the course in order to follow the path.

7.2.2 LOS based approach

The approach presented in the coming section utilized LOS guidance during all maneuvers. Where one or more intermediate waypoints are created between the global waypoints if tacking or jibing is needed, this is illustrated in figure 7.9. The approach is hence conceptual different from the approach in section 7.2.1, as the algorithm here need to plan a path when crossing is required. Despite the difference, the fundamental calculations are same.

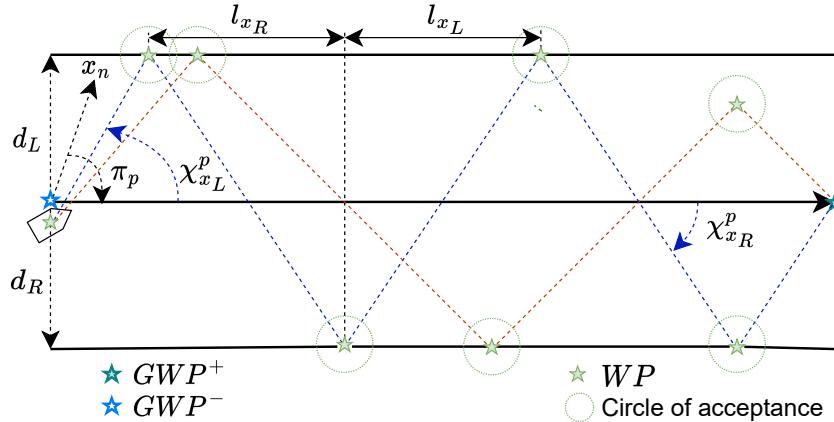


Figure 7.9: LOS based crossing path

The approach presented in the coming section utilized LOS guidance during all maneuvers, where one or more intermediate waypoints are created between the global waypoints if tacking or jibing is needed, as illustrated in figure 7.9. The approach is hence conceptual different from the approach in section 7.2.1, as the algorithm here needs to plan a path when crossing is required. Despite the difference, the fundamental calculations are the same.

Figure 7.9 shows a crossing path where LOS is utilized. The path width is constrained by d_L and d_R , which represent the line where the local waypoints are located. The along-path progress length for crossing left and right are denoted l_{x_L} and l_{x_R} respectively. While $\chi_{x_L}^p$ and $\chi_{x_R}^p$ are the crossing courses given in the path frame.

The algorithm is not covered in dept, but the basic working concept is described here and illustrated in figure 7.9:

1. Compute the path angle π_p and the cross-track error from the path given by GWP^+ and GWP^- .
2. Use the path angle, ocean current and wind to determine if crossing is required based on the material established in section 7.1. If required, the optimal crossing angles are found using the same material.
3. Rotate the course angles into the path frame such that one can distinguish between left and right of the path.
4. If crossing is needed: Plan intermediate waypoints by utilizing the optimal crossing courses, path width and the resulting along-path progress for every straight-line segment. As l_{x_L} and l_{x_R} will be periodic, the location of the first local WP gives the location of the rest. The only special cases to treat separately are the global path segment's start and end. Further, the local waypoints are updated.

5. Execute path using LOS guidance.

In this algorithm, the straight-line segments are changed directly, with no circular path following in between the segments. This is a simplification compared to reference [3], which uses LOS to follow an inscribed circle between the straight-line segments. Despite the simplification, the LOS guidance law was observed to make reasonable maneuvers in simulation. Notice that this algorithm either needs to update the list of global waypoints or have a separate waypoint switching mechanism for the local path. The latter is used in this thesis.

7.3 Path manager

The GWP_s are assumed to be predefined by a user or a route optimization layer. Hence the path manager only consists of a simple waypoint switching mechanism, which uses a circle of acceptance. Here the algorithm updates the waypoints when the craft is within a radius of acceptance relative to the target waypoint on the current path. In other words, the waypoints are updated when the following relation is satisfied:

$$\sqrt{(x_{i+1}^n - x^n)^2 + (y_{i+1}^n - y^n)^2} \leq R_{acc} \quad (7.22)$$

(7.23)

Where R_{acc} is the acceptance radius, and the x_{i+1}^n and y_{i+1}^n are the coordinates of GWP^+ .

The same method is used to switch the local waypoint for the LOS-based path following approach.

7.4 Collision avoidance

In this section, the collision avoidance formulation is presented, together with the applied simplifications and assumptions. The solution forms the mid layer of the guidance system that provides an alternative path if an obstacle blocks the path between two global waypoints.

In reference, [11] interval analysis is used to select between progressing on the path or do station keeping. In reference [13] courses towards obstacles are penalized based on distance, resulting in a minimization problem. Within that work, a modified FVD is used to represent the penalty. In addition, the crossing regions are considered non-optimal, meaning only straight courses are evaluated. While in reference [14] a potential field algorithm is used to make the target position attractive and incorporate COLAV. Within this thesis, a simple COLAV algorithm for static environments are developed using the velocity obstacle. The principle of the method is based on relative velocities and can hence be further developed to be used in dynamic environments. For the case where all paths are blocked, the saildrone is assumed to do station keeping either by sailing in an 8-shape to preserve controllability or use the backup propulsion system. The method applied here takes into account that crossing might be needed during the collision avoidance maneuver as wind direction might change. Hence the collision-free paths taken are very conservative if crossing turns out not to be needed; however, it enables to select any desired collision-free course.

The formulation utilizes the velocity obstacle and perception discussed in section 2.7 to separate the surrounding environment into velocity obstacles and free cones, where a collision-free course can be selected within the free cones. The Velocity obstacles also take into account the width of the resulting path, as tacking or jibing might be required to reach the target position. As no prior knowledge of the obstacles is available, a course is selected within the free cones using a greedy choice. Taking the course that deviates the least from the original path seen from the current position in $\{n\}$.

The COLAV module outputs intermediate waypoints when the path is blocked, where the waypoints are passed to the path following module. When the direct direction to the target becomes unblocked, the previous global waypoint is overwritten by the current position, resulting in a new path segment.

Expanding the obstacles

For every position in $\{n\}$ to be reachable, tacking or jibing will be required, which means that these maneuvers need to be incorporated into the velocity obstacle. This is done by expanding the obstacle with half of the width of the worst-case zigzagging path, assuming the resulting path is centered. This is done for all paths

regardless if tacking/jibing is needed or not, as the wind direction might change underway.

In addition, the obstacles are expanded to incorporate the dimensions of the saildrone. As the saildrone's maneuverability is limited, it is introduced as a circular shape with a diameter equal to the length of the hull. This implies that all obstacles must be expanded by half of the hull length. This simplifies the COLAV problem as the saildrone's orientation does not change the expansion of the obstacles. Further, a safety margin is added to the expansion to give some margins for maneuvering errors, disturbances, etc. In general, this safety margin is large for practical implementation. However, a smaller margin is applied in the simulations to show the concept. The total expansion distance is given by:

$$d_{expand} = \frac{l_{hull}}{2} + \frac{dR + dL}{2} + d_{safe} \quad (7.24)$$

The expansion leads to a set of velocity obstacles, as shown in figure 7.10, where all velocity vectors not contained within these VO cones result in collision-free paths. Within a static environment with no moving obstacles and no ocean current, the cones are centered at the saildrone. Hence every allowable velocity vector corresponds to a saildrone course. The collision avoidance problem is hence solved by selecting a course that is not within any of the velocity obstacles $\pi_p \notin VO$, but rather is contained in the collision-free cones FC . Notice that path angle π_p is used as the resulting course will converge to the path angle. This also allows tacking and jibing to be applied to achieve the desired path angle.

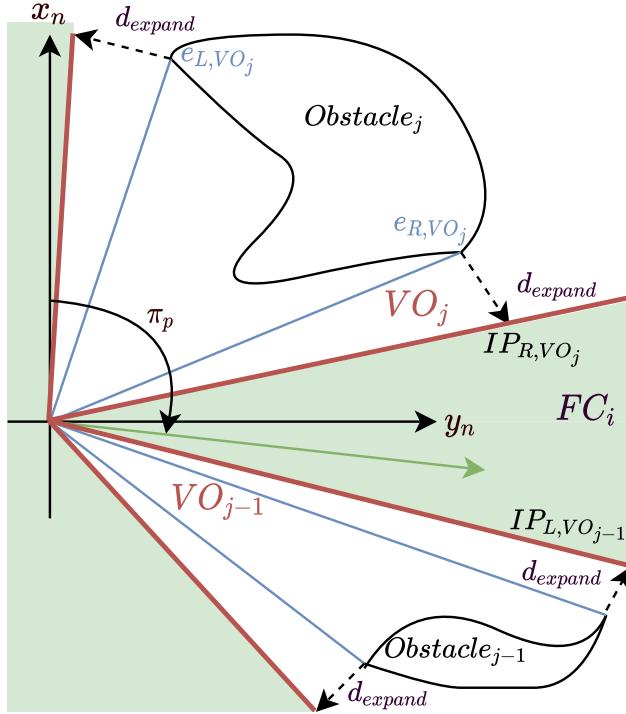


Figure 7.10: Velocity obstacles and free cones

Figure 7.10 show the velocity obstacles (VO) and the collision-free cones (FC). Together with the expansion distance d_{expand} which is perpendicular to the vector form CO that intersects the outer contour of the obstacle (e_L and e_R). While IP are virtual intersection points created by the triangle of the intersection vector and d_{expand} .

Path course compensation

Figure 7.11 shows how the VOs change when utilizing tacking or jibing. The change is caused by the cross-track error y_e^P from the path given by the two intermediate waypoints WP^- and WP^+ . This implies that the course π_i given by the intersection point IP needs to be compensated by an angle χ_{offset} , to maintain a path angle equal to or similar to the previous path given by WP^- and WP^+ . The compensation is needed as the new perspective indicates that the original course is within the VO; however, the resulting course is in fact not in the VO as seen from the path frame $\{p\}$.

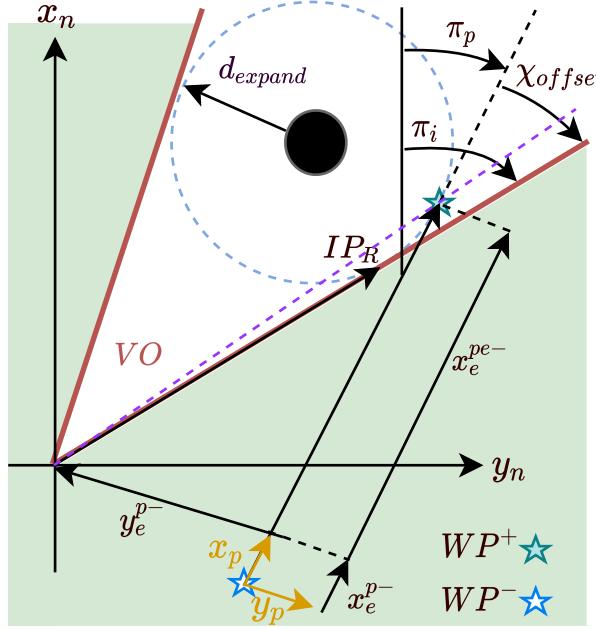


Figure 7.11: VO when tacking or jibing

A simple approximation of the compensation angle is given by equation 7.25 using the cross-track- y_e^{p-} and along-track error x_e^{pe-} with respect to the previous set of waypoints (WP^- and WP^+) seen in figure 7.11. This is reasonable when the evaluation intervals are small relative to the dynamics of the saildrone and the environment, such that the intersection point remains close to WP^+ at every iteration. As indicated by figure 7.11, the approximation in equation 7.25 given by the purple line will often give a conservative path angle π_p with respect to the VO when the cross-track error is non zero. Also notice that the next WP^+ should be selected such that $x_e^{pe-} > 0$.

$$\chi_{offset} \approx -\tan^{-1}\left(\frac{y_e^{p-}}{x_e^{pe-}}\right) \quad (7.25)$$

The updated path angle π_p becomes

$$\pi_p = \pi_i - \chi_{offset} \quad (7.26)$$

Notice the similarity to the lookahead-based LOS guidance law. If one solves for π_i the expressions are identical where the lookahead distance equals the along-track error up to the next waypoint.

The following updates are used for the intermediate waypoints:

$$WP^+ = IP \quad (7.27)$$

$$WP^- = WP^+ - a[\cos(\pi_p); \sin(\pi_p)] \quad (7.28)$$

Where a is a positive constant.

COLAV algorithm

The collision avoidance algorithm in algorithm 2 is obtained by using the definitions from figure 7.11 together with equation 7.24-7.28. The algorithm can be described as follows: Given the global path consisting of the next and previous waypoints GWP^+ and GWP^- . If the path from the current position to the target (GWP^+) is blocked by an obstacle, select an intermediate path consisting of 2 waypoints WP^+ and WP^- . Where the waypoints are given by the path to the left or right of the velocity obstacle(s) that deviate the least from the original path. When left or right is selected, the decision is maintained until the direct path to the target is no longer blocked. The waypoints are adjusted at every iteration of the algorithm, using equation 7.27 and 7.28. When the path is not blocked anymore, the previous global waypoint is overwritten by the current position, resulting in a new global path segment towards the target.

Notice that the algorithm can adjust both the path angle and the path position when waypoints are used as output. Allowing the path to always run through the intersection point IP independent of if tacking or jibing is needed. Remark that this solution will lead to jumps in the cross-track error when the waypoints are updated. However, the effect will be small when the cross-track error is small, as the compensation term will be close to zero, meaning the path angle is mainly adjusted when the waypoints are updated. This property is important as LOS guidance uses the cross-track error to determine the desired course. However, when tacking or jibing, the cross-track error from the path can be high. In this case, the path following module will use pure course control, meaning the cross-track error will not affect the desired course.

The variables and states used in algorithm 2 is given in table 7.2. Where bc and bc_{max} are used to avoid fluctuation when the obstacle is at the border of the detection range. While $ps = -1$ indicates that the saildrone will go to the left of the obstacle. $ps = 1$ is to the right, while $ps = 0$ means that right or left has not been decided yet. Notice that bc and ps are stored and used in the next iteration. This is also the case for the previous global waypoint when it is updated.

Table 7.2: State and variable definitions algorithm 2

Description	Notation
Indicate if a alternative path is needed	ap
Number of times the path has been blocked in a row	bc
Number of times the path has to been blocked in order for the COLAV to start	bc_{max}
Previous decision state	$ps \in [-1, 0, 1]$
Intersection distance given by magnitude of IP	ID

Algorithm 2 VO based COLAV

Input: GWP^+ , GWP^- , VO, x^n , y^n , x_e^{pe-} , y_e^{p-} , bc_{max}

Output: WP^+ , WP^-

Initialize: ps, bc

$$\pi_{Gp} \leftarrow \text{atan2}(GWP^+(2)-GWP^-(2), GWP^+(1)-GWP^-(1)) \quad \triangleright \text{Angle of global path}$$

$$\chi_t \leftarrow \text{atan2}(GWP^+(2)-y^n, GWP^+(1)-x^n) \quad \triangleright \text{Direct course to target}$$

$$ap \leftarrow 0$$

```

if  $\chi_t \in VO$  then
     $ap \leftarrow 1$ 
     $[\chi_{Rvo}, \chi_{Lvo}] \leftarrow \text{get collision free course}$ 
     $[ID_R, ID_L] \leftarrow \text{get intersection distance}$ 
    if ps == 0 then
        Select collision free course closest to  $\pi_{Gp}$ , ps  $\in \{-1, 1\}$ 
    end if
    if ps == 1 then
         $\pi_i \leftarrow \chi_{Rvo}$ 
         $ID \leftarrow ID_R$ 
    else
         $\pi_i \leftarrow \chi_{Lvo}$ 
         $ID \leftarrow ID_L$ 
    end if
end if

if ap == 0 then
    if bc ==  $bc_{max}$  then  $\triangleright$  The path to target has been blocked before
         $GWP^- \leftarrow [x^n; y^n]$   $\triangleright$  Update  $GWP^-$  for next iterations
    end if
    ps  $\leftarrow 0$ 
    bc  $\leftarrow 0$ 
else
    bc  $\leftarrow bc + 1$   $\triangleright bc \in [0, bc_{max}]$ 
end if

if ap == 1 and bc ==  $bc_{max}$  then
     $\chi_{offset} \leftarrow -\text{atan}(y_e^{p-}/x_e^{pe-})$   $\triangleright x_e^{pe-} > 0$ 
     $\pi_p \leftarrow \pi_i - \chi_{offset}$ 
     $WP^+ \leftarrow [ID \cos(\pi_i) + x^n; ID \sin(\pi_i) + y_n]$ 
     $WP^- \leftarrow WP^+ - 100[\cos(\pi_p); \sin(\pi_p)]$ 
else
     $WP^+ \leftarrow GWP^+$ 
     $WP^- \leftarrow GWP^-$ 
end if

```

7.5 Guidance system parameters

The main parameters used in the guidance system are summarized in table 7.3. The selected look-ahead distance was observed to work well across simulations and also when using LOS for crossing. The detection range and safety distance are kept relatively short for the simulation cases.

Table 7.3: Guidance system key parameters

Subsystem	Description	Symbol	Value	Unit
Path following	Look-ahead distance	Δ	40	m
	Frequency		10	Hz
COLAV	Detection range		300	m
	Safety distance	d_{safe}	20	m
	Frequency		0.2	Hz
Path manager	Circle of acceptance	R_{acc}	20	m
	Frequency		1	Hz

Chapter 8

Simulations results

In this chapter, different simulation cases are studied in order to prove the working concept of the motion control system as a whole and to demonstrate individual features. In addition, the performance and robustness gained by exploiting ocean current in the CVD are studied. The two approaches to path following are also compared both in terms of obtained velocity and margin to the dead zone. At the end of the chapter, a long-mission simulation is studied, along with a set of collision avoidance cases.

For the simulations, the framework presented in chapter 5 is utilized, which consist of a simplified perception model together with the simulation model developed through section 4.1.

In the simulations, the wind velocity is within the range $U_w \in [4, 16]$ m/s, which is light to moderate breeze. The ocean current velocity is within $U_c \in [0, 1]$ m/s, which is the same range used to visualize the ocean surface velocities in [44]. Note that the wind influences the ocean current; however, the current is also subjected to other contributing terms, such as the Coriolis effect and tidal current [45]. This means that the wind and current velocity vector does not need to be parallel.

8.1 Path following comparison

In this section, crossing using intermediate waypoints in combination with LOS guidance is compared to crossing using the course and LOS altering approach. To simplify the writing, the two methods are referred to as LOS-based and altering-based. In the following section, current will not be accounted for in the CVD in order to use the ocean current as a disturbance. Notice that the LOS-based approach is only used in this section. For the rest of the thesis, the path following algorithm used and referred to is the course and LOS altering approach.

The main difference between the two methods is that: The altering-based method maintains a path by periodic switching from one desired course to another when executing crossing. This allows the saildrone to maintain the optimal course independent of disturbances and progression along the global path. While the LOS-based method plans and creates a local path that consists of intermediate waypoints between the global waypoints. Such that LOS guidance can be used under all maneuvers.

As earlier established, the two methods utilize the same desired crossing courses. Hence the path should be identical in an ideal situation. However, the principle when the local path is executed is different as LOS uses the course angle to reduce the cross-track error, which is a drawback as the optimal course can be compromised. In addition, with high cross-track error, there is a risk of going too steep into the wind, as the margin to the dead zone is reduced when "falling behind" the local path. This is illustrated in figure 8.1, where one also can see that the margin increases if the saildrone is in "front" of the path. The latter case can be achieved by switching waypoints late in order to overshoot the path. However, this compromises the course as well, but only for a limited time as the saildrone converges to the path. Note that when the cross-track error is zero, the course will be optimal.

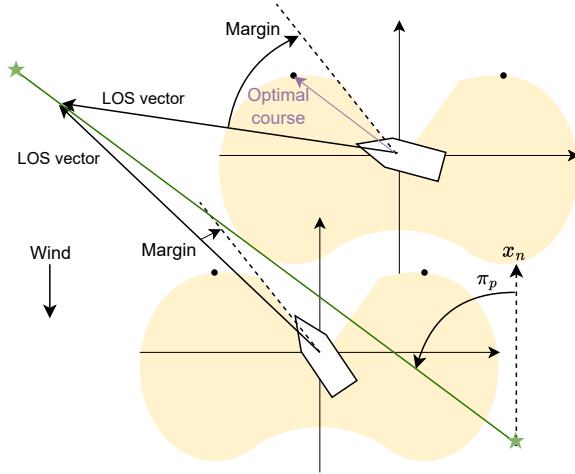


Figure 8.1: Margin to dead zone when using LOS for crossing

A simulation study where the LOS-based method is used is shown in figure 8.2, where late and early waypoint switching is compared. This result in over- and under-shooting of the planned path. In figure 8.2a one can observe the reduced forward propulsion when under-shooting the path. The saildrone tries to converge back to the path, but the course is too steep for the sail to provide sufficient propulsion. Hence both the velocity along the local and global path is reduced. The loss of control at $x^n = 200$ m is caused by the reduced velocity combined with an

early WP switching using tacking. This would have been prevented if an upwind jib had been used instead; however, this was not implemented in the LOS-based solution. Overshoot results in a course that provides higher propulsion at the exit of the tacking maneuvers, as indicated in figure 8.1. At the same time, this means that the maneuver requires a longer turn; hence there will be a trade-off between the length of the turn and the exit propulsion.

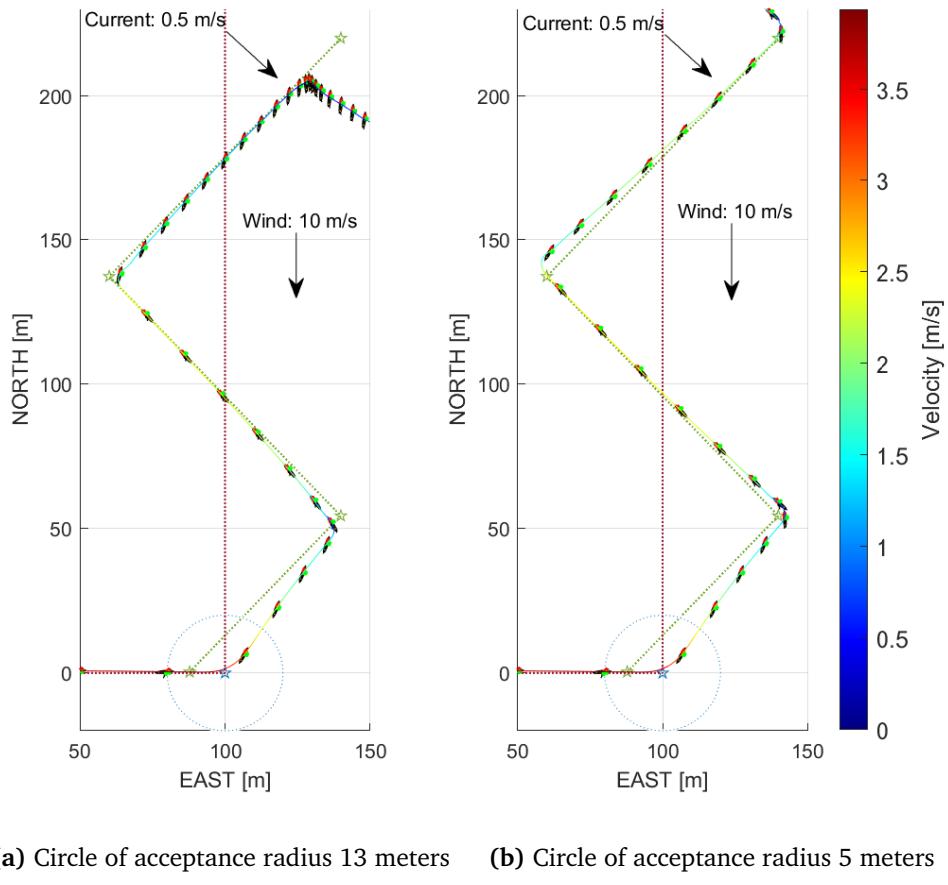


Figure 8.2: Crossing using LOS-based path following

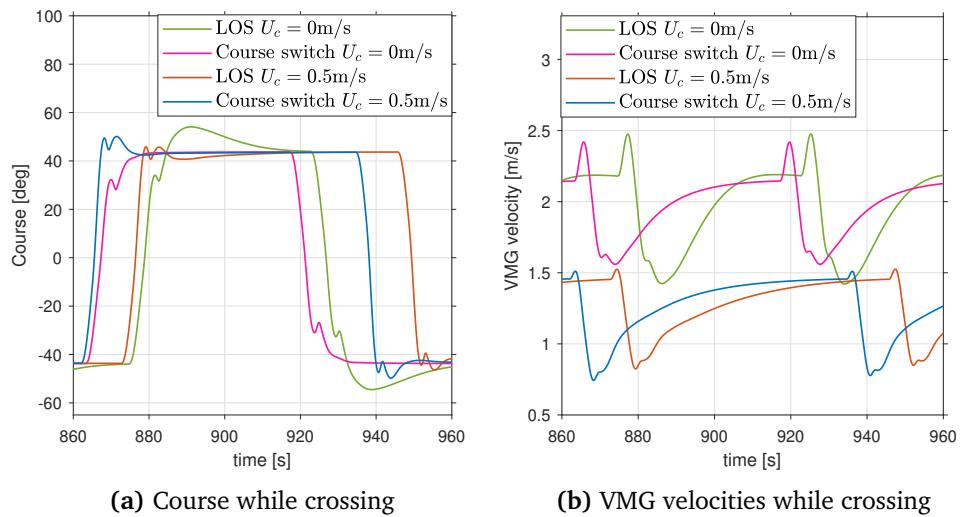
A comparison race of the two approaches

To see the performance difference between the two methods, the altering-based method was compared to the LOS solution through a 2 km simulation race with a path directly against the wind. Two tests were conducted, the first without current and the second with 0.5 m/s current along the wind direction. The path width was adjusted in order to make these as similar as possible. The obtained finish times can be seen in table 8.1.

Table 8.1: Course switching and LOS based path planning finish times

Wind velocity [m/s]	10	10
Current velocity [m/s]	0	0.5
Finish time course switching [s]	1044.2	1577.2
Finish time LOS [s]	1058.2	1588.2

The results show that the altering-based method performs better than the LOS approach for the selected configuration of the two methods. The performance is not significantly different but noticeable over long-term missions. From figure 8.3a, it can be seen that the LOS-based method has a course overshoot of approximately 10° when the current is zero. Despite the higher velocity the overshoot gives when exiting the tacking maneuver, the resulting VMG velocity is lower. This can also be seen from figure 8.3b, where the altering-based method in general holds higher VMG velocity in longer periods than with LOS.

**Figure 8.3:** Course and VMG velocity for conducted comparison

Considering the finish times with ocean current, it is clear that the difference is reduced, which is caused by the ocean current and its direction. For the LOS approach, the current helps reduce the overshoot and hence the cross-track error, as shown in figure 8.3a. The course comes closer to the optimal course faster; despite this, the saildrone ends up "behind" the path with a course below the optimal. This indicates that trying to overshoot the path does not fully resolve the drawback LOS imposes. One can also observe that the current creates an overshoot for the altering-based method, when the current switches from one side to the other. Figure 8.3b shows that the course overshoot results in a lower VMG velocity at the tack exit. However, the fast convergence to the optimum result in a better average VMG for the altering-based method, leading to a better finish time.

From these tests, one can conclude that similar or equal performance can be obtained by these two methods. However, the performance of LOS-based approaches will vary more with disturbances and hence require more parameters to be adjusted to follow the optimal path with a similar performance as the other method. If one only considered the straight line segments, it is clear that the LOS never will perform better than a course controller (assuming both a perfectly tuned), as the course controller will always try to hold the optimal course angle. When considering the material presented in section 7.1 it is clear that only an effective course towards the target destination is needed, meaning there is in fact not an optimal path to follow but rather a set of courses to maintain. This means a course controller is desirable. In a dynamic environment, wind, current and other disturbances will change such that the planned path becomes less optimal, also indicating that a course controller might be desirable.

8.2 Wind and ocean current compensation when crossing

In this section, the path following algorithm is subjected to different wind and ocean current conditions to show how these environmental conditions change the crossing pattern. Proving the working concept of the algorithm through simulation. Under these simulations, a path width of 60 meters is used.

8.2.1 Wind influence

The influence of the wind direction when crossing is required can be seen in figure 8.4. The path following algorithm adjusts the optimal course according to section 7.1, while the path width is constrained to a finite width. As the constraints are only used to switch course, they do not represent the outer boundaries of the path. As seen in figure 8.4b, at $x^n = 185$ meters, the course, saildrone velocity and heading reference model make the saildrone travel further from the path than in figure 8.4a. This indicates that a safety margin should be added if the system must comply with strict boundaries.

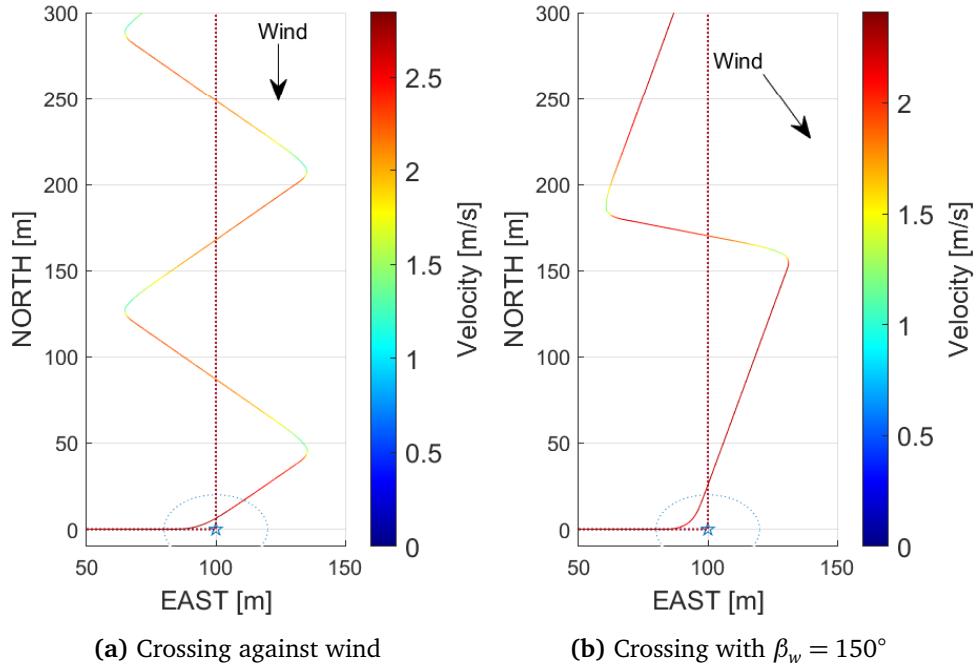


Figure 8.4: Crossing with 8 m/s wind

8.2.2 Ocean current influence

The combined influence of ocean current and wind is shown in figure 8.5, where the nominal path without current from figure 8.4a is the dashed line. From figure 8.5a it is clear the path following algorithm selects a less step course due to the shifting effect the current has on the CVD. Notice that this helps preserve the margin towards the dead zone, which from the perspective of the saildrone's velocity in $\{n\}$ now will contain a larger set of course angles due to the shift of the CVD. When current enters from the side, as shown in figure 8.5b, the forces induced by the keel and hull make it possible to go closer against the wind during the crossing course going into the current. While the opposite happens when crossing towards the right along the current.

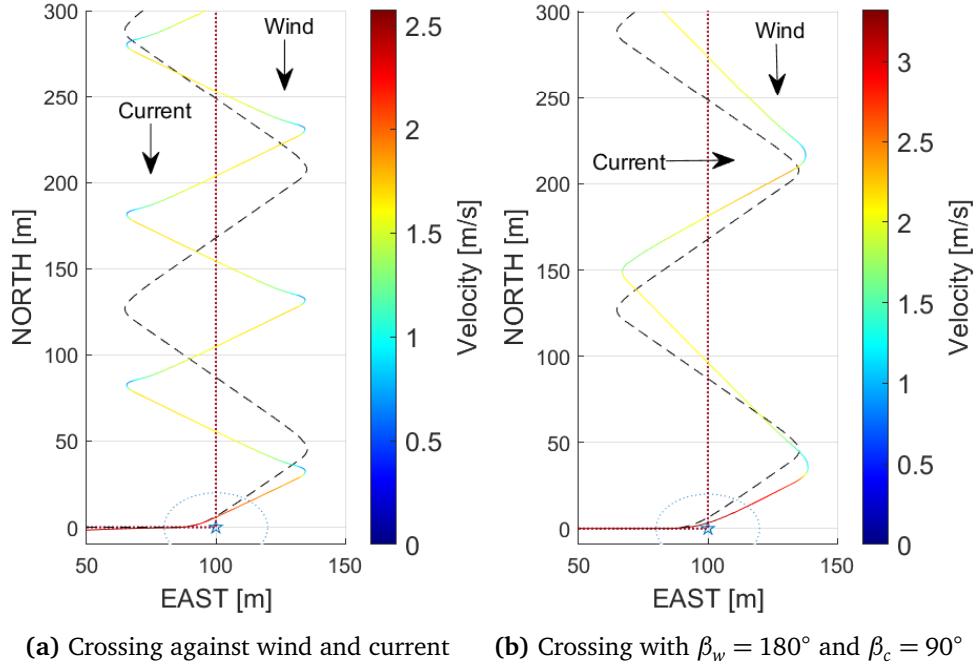


Figure 8.5: Crossing with 8 m/s wind and 0.5 m/s ocean current

8.3 Jibing and tacking simulation

Here a global path is simulated using both tacking and upwind jibing. To maintain the same conditions under the simulations, the upwind jibing threshold is increased to 3 m/s to prevent the saildrone from tacking in one of the simulations. The path width is selected to 80 meters, wind at 8 m/s, and an ocean current of 0.3 m/s. The wind direction is set to $\beta_w = 180^\circ$ while the current direction is $\beta_c = 150^\circ$.

In figure 8.6 tacking is used upwind, while jibing downwind. From the figure, it can be concluded that the path following algorithm performs well under waypoint switching. At the waypoint $(x^n, y^n) = (150m, 100m)$, a small course change can be seen when the WP is reached, caused by the fact that WP^- has been updated by the algorithm. In the last tack before the WP, it detects that the course to the WP is feasible without crossing; hence WP^- is updated and the saildrone progress further using LOS. When the WP is reached, the next path requires crossing, and the optimal crossing course is used. At waypoint $(x^n, y^n) = (250m, 150m)$, it can be seen that the crossing lead to an undesirable maneuver, where the saildrone has to travel back to the WP. This effect will be neglectable for larger missions with longer distances between the GWP.

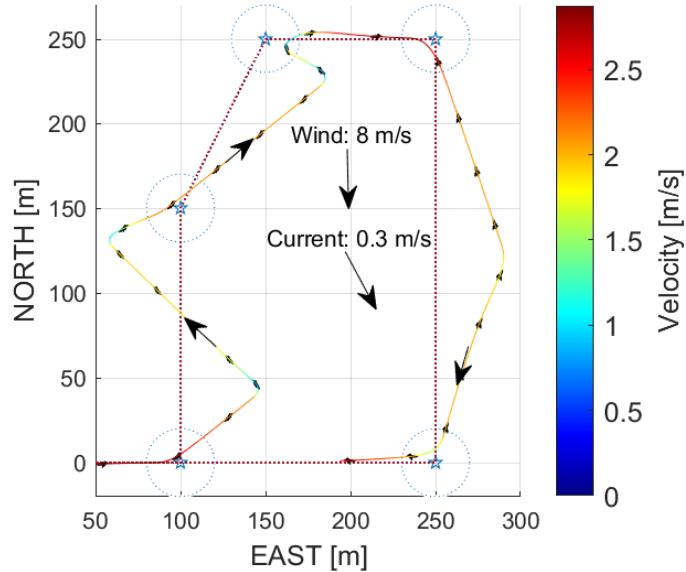


Figure 8.6: Upwind tacking and downwind jibing

Both upwind and downwind jibing can be seen in figure 8.7. Also in this case the switching between waypoints is reasonable. The deformation of the upwind jibing loops is a result of the current. However, this was not observed to cause stability issues. This is expected as the algorithm will only use course control during jibing, making it independent of the global path cross-track error.

Upwind jibing within the same constraints as for tacking yields significantly slower progress along the path. This is caused by both the loops and the resulting shorter straight line segments, which reduce effective progress and obtainable velocity along the path. In addition, the number of maneuvers is increased. For comparison, the path to follow in figure 8.7 and 8.6 are the same. However the spent time until $(x^n, y^n) = (0m, 200m)$ is reached, is 800 seconds with upwind jibing, while 450 seconds with tacking. The path width is adjustable; hence faster progress can be obtained if a wider path is allowed.

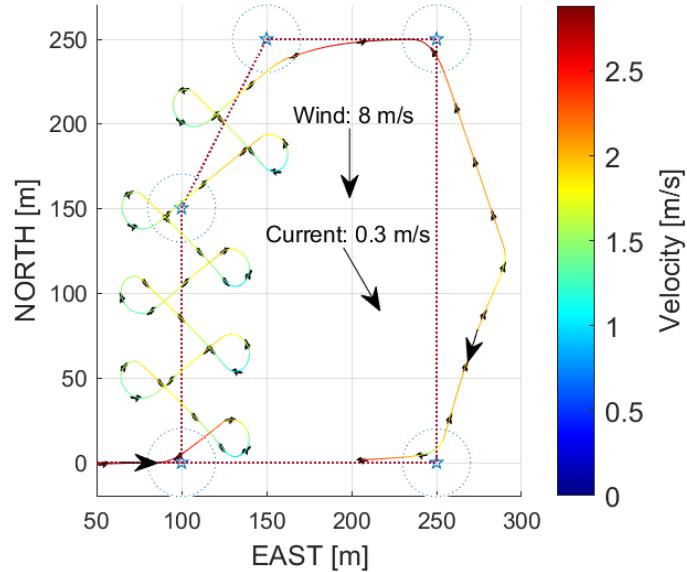


Figure 8.7: Upwind and downwind jibing

8.4 Improvement by including ocean current

In the following simulation cases, the path following algorithm is tested with and without accounting for current in the CVD to reveal the potential differences.

Both simulation cases consider a 1 km long path going directly into the wind. The first case considers ocean current acting along the wind, while the second considers ocean current acting normal to the path. The goal is to compare the time spent and velocities obtained during continuous crossing for different ocean current conditions. Hence for both simulation cases, the saildrone is executing crossing before the start line to have stable crossing along the whole path. The improvement in average VMG velocity given in table 8.3 and 8.5 are with respect to the same test conducted without accounting for the ocean current.

8.4.1 Sailing against the wind and ocean current

The first simulation case is a comparison race, where $\pi_p = 0^\circ$, $\beta_c = 180^\circ$ and $\beta_w = 180^\circ$. A set of different wind and current velocities was tested, as shown in table 8.2 and 8.3. As shown in the latter table, including the current into the course selection has significant potential at low to medium saildrone velocity. This is expected when considering section 7.1, as at lower wind velocity, the CVD is small, meaning the feasible velocities are low. Hence, the current's contribution

to scaling, rotation and shifting of the CVD becomes important. The same effect is reflected in the simulation cases. At 8 m/s wind and 0.5 m/s current, the increase in VMG velocity is 10.4 %. However, when the wind is increased to 12 m/s, the effect is neglectable.

The results obtained are relevant for any sailboat at calm wind due to the constrained velocity. However, when the wind and CVD allow higher velocities, the effect is neglectable, as one can see in the tables as the wind velocity increases. This also implies that for low and mid-range speed sailboats like the design considered in this thesis, the presented course compensation will give higher velocity when crossing. In addition, the compensation maintains the margin to the dead zone and hence provides robustness. For the last configuration in table 8.2, the saildrone did not finish as the path was too steep against the wind and current, making the saildrone lose control. This understates the robustness introduced, as this was not an issue when accounting for ocean current.

Table 8.2: Velocities and finish times for 1 km path against the wind and current, without accounting for current.

Wind velocity	Current velocity	Finish time	Max velocity	VMG velocity
8 m/s	0.5 m/s	1297.7 s	1.15 m/s	0.77 m/s
10	0.5	774.8	1.98	1.29
12	0.5	565.8	2.72	1.77
12	0.8	781.0	2.01	1.28
12	1.0	-	-	-

Table 8.3: Velocities and finish times for 1 km path against the wind and current, while accounting for current.

Wind velocity	Current velocity	Finish time	Max velocity	VMG velocity	VMG increase
8 m/s	0.5 m/s	1174.9 s	1.70 m/s	0.85 m/s	10.4 %
10	0.5	753.7	2.37	1.33	3.1 %
12	0.5	562.5	3.08	1.78	0.6 %
12	0.8	708.5	2.76	1.41	10.2 %
12	1.0	864.6	2.52	1.16	-

8.4.2 Sailing against the wind with ocean current from the side

The second simulation case is a comparison race, where $\pi_p = 0^\circ$, $\beta_c = 90^\circ$ and $\beta_w = 180^\circ$. The case is illustrated in figure 8.9. The conducted simulations are summarized in table 8.4 and 8.5. As seen in table 8.5 the increase in VMG velocity is even higher with ocean current from the side. However, increasing the

feasible velocity of the saildrone reduce the effect also here.

Now considering the case when $U_c = 1 \text{ m/s}$ and $U_w = 12 \text{ m/s}$, which is shown in figure 8.9, where the difference in VMG velocity is 31.1 %. This is an extreme case, but it illustrates the cause of the performance difference well. The uncompensated course for crossing (meaning current is not accounted for) towards the right requires a higher keel AOA compared to the compensated course as indicated by both figure 8.9 and 8.8. The high keel AOA combined with the resulting small angle between the path and the saildrone heading gives a large force component towards the left. Which acts as a brake when the saildrone's velocity component parallel to the current is larger than the ocean current. The result can be seen in figure 8.9a, where the saildrone velocity decay when going toward the right. The resulting low relative velocity with respect to the ocean gives controllability issues, as seen in figure 8.8 in the range $t \in [230, 260]\text{s}$, where the rudder AOA oscillates and the rudder quickly stalls. This was also the reason why the saildrone lost control under the test at 6 m/s wind, where the low relative velocity with respect to the ocean current gave neglectable rudder forces. However, this issue is prevented when incorporating the current, as shown in figure 8.9b. Because a higher course angle is applied, reducing the keel AOA. The increased velocity when going right also leads to a smoother tacking maneuver, where less energy is lost. When going left the current allows a course closer to the wind. As seen from figure 8.9 the saildrone velocity is lower compared to the uncompensated course, but the VMG velocity is higher.

Table 8.4: Velocities and finish times for 1 km path against the wind and $\beta_c = 90^\circ$, without accounting for current.

Wind velocity	Current velocity	Finish time	Max left/right velocity	VMG velocity
6 m/s	0.5 m/s	-	-	-
8	0.5	777.4 s	2.25, 1.61 m/s	1.29 m/s
10	0.5	531.9	2.97, 2.62	1.88
12	0.5	428.5	3.61, 3.34	2.33
12	1.0	864.6	3.52, 2.01	1.86

Table 8.5: Velocities and finish times for 1 km path against the wind and $\beta_c = 90^\circ$, while accounting for current.

Wind velocity	Current velocity	Finish time	Max left/right velocity	VMG velocity	VMG increase
6 m/s	0.5 m/s	1010 s	1.39, 1.42 m/s	0.99 m/s	-
8	0.5	658.3	2.07, 2.40	1.52	17.8%
10	0.5	500.7	2.80, 3.07	2.00	6.3 %
12	0.5	409.8	3.40, 3.77	2.44	4.7 %
12	1.0	406.0	3.20, 3.85	2.46	31.1%

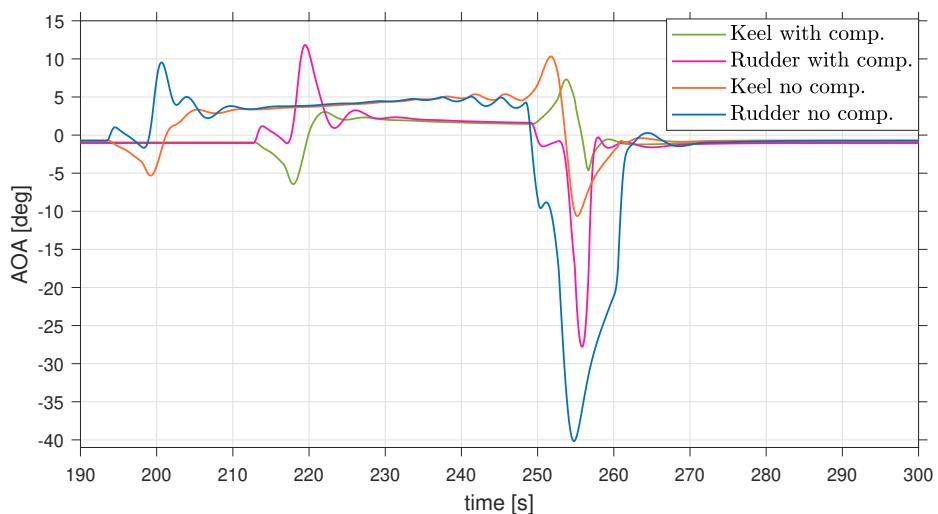


Figure 8.8: Keel and rudder AOA during tacking at 12 m/s wind and 1 m/s current

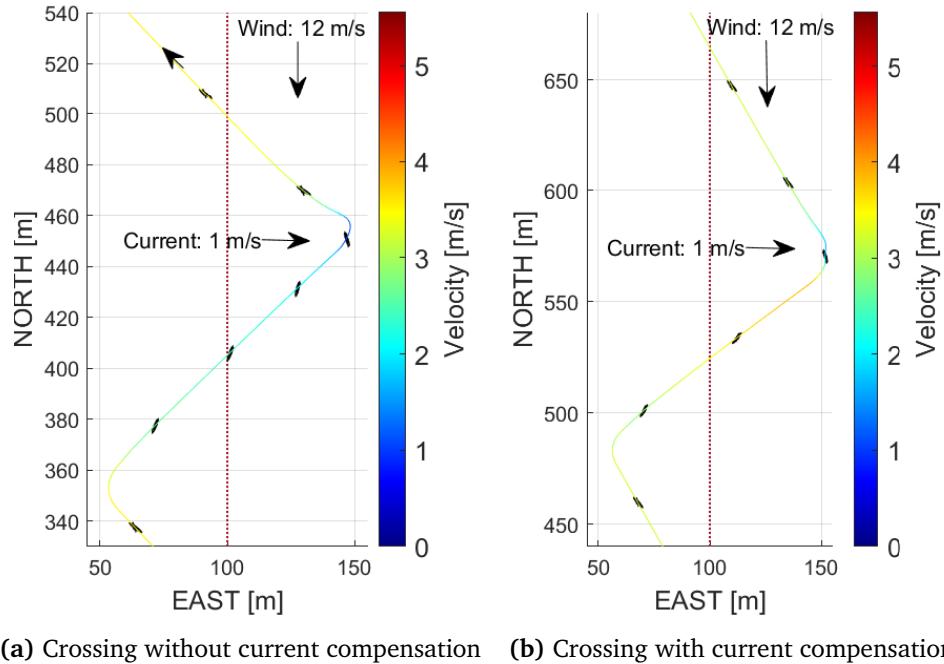


Figure 8.9: Crossing with and without considering ocean current contribution

8.5 Long mission simulation

In this section, a long mission simulation is presented together with the resulting evolution of the system states. Both the wind and current direction is set to 180° , where the current velocity is 0.1 m/s . While the wind changes from 6 m/s to 10 m/s over a period of 50 seconds.

The resulting path and velocities can be seen in figure 8.10, where the increase in wind magnitude enables the saildrone to go from upwind jibing to tacking. One can also observe that the transition between the path segments has varying path overshoot, dependent on the saildrone velocity, wind and current, together with the angle of the previous and next path segment. However, this is expected as the circle of acceptance and look-ahead distance are constant. Despite the varying path overshoot, the saildrone prevents this under crossing, due to the use of course control instead of LOS.

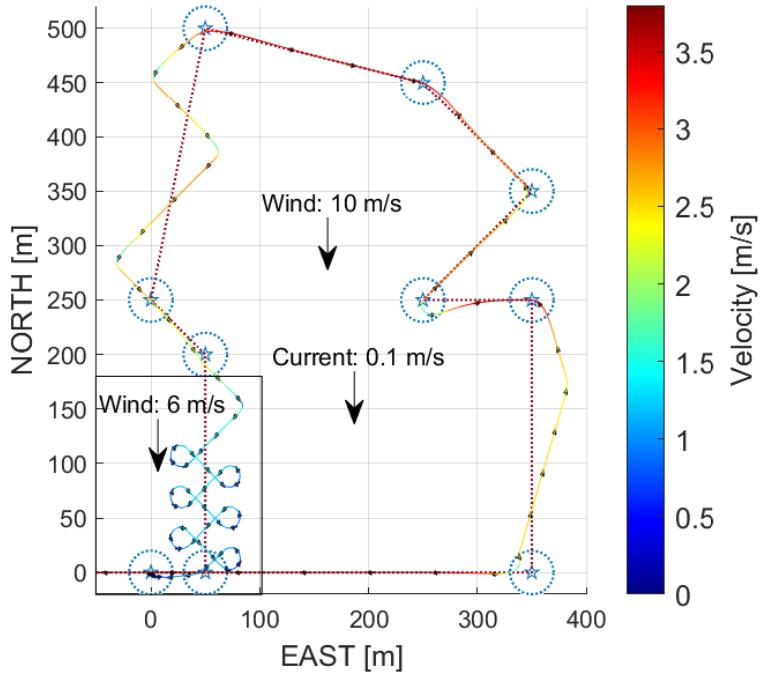


Figure 8.10: Long mission under charging wind magnitude

The system states for a section of the mission is shown in figure 8.11. These are the essential states related to control of the saildrone's heading and the sail AOA. In addition, the roll angle is shown, as the roll dynamics introduce disturbance into the control systems. Despite the disturbance from the roll dynamics and other effects, the rudder and tail angle have stable behaviors. Leading to a desirable transient response of the sail AOA and heading. Also, notice that the sail AOA remains stable at high AOA.

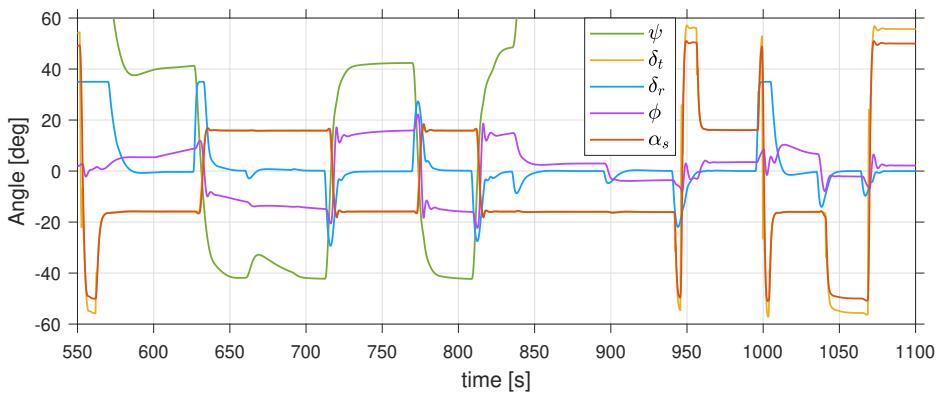


Figure 8.11: States during long mission

Considering the response in the range $t \in [925, 1100]$, it can be observed that the sail AOA changes between 4 optimal angles when the saildrone need to change between two courses that pass across the dead run zone. This is not a problem when the sail system dynamics is faster than the saildrone heading dynamics, such that the optimal angle can be reached. However, when this is not the case, it might be desirable to skip using drag-mode when passing through the dead run zone.

8.6 Collision avoidance simulation

In this section, a set of collision avoidance cases are studied for a static environment with no ocean current. A simple circular obstacle with a diameter 40 meters is used, along with a safety margin of 20 meter. The detection range is set to 300 meters, while the crossing path width is 60 meters. In case 1 to 3 the direction of the wind is set to $\beta_w = 180^\circ$, while in case 4 it is $\beta_w = 150^\circ$. The simulation cases are shown in figure 8.12 and 8.13, where the light blue circle indicate the expanded obstacle. Meaning the resulting collision-free course will be the tangent to the circle and the saildrone is allowed to be inside it. The dark blue circle is the obstacle, together with the safety margin and the extent of the saildrone. Which represents the region the saildrone should not enter.

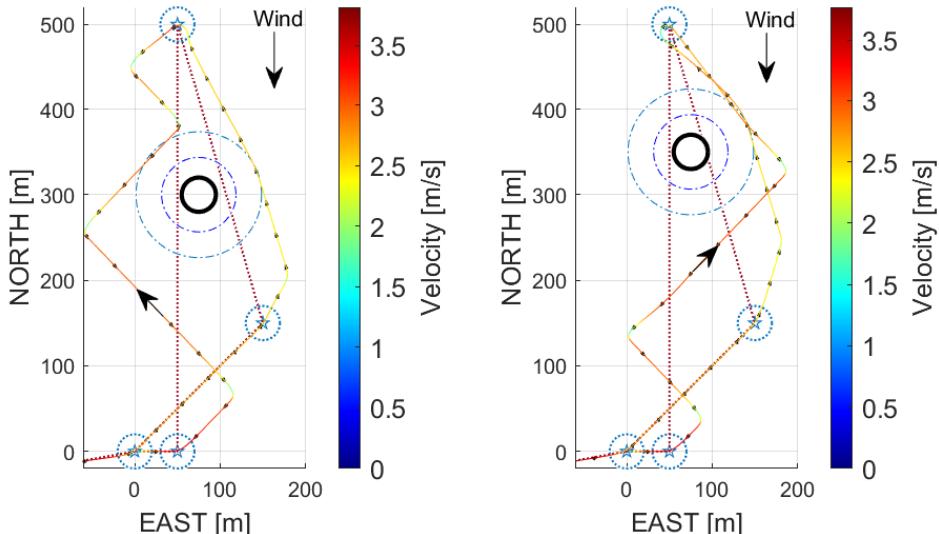
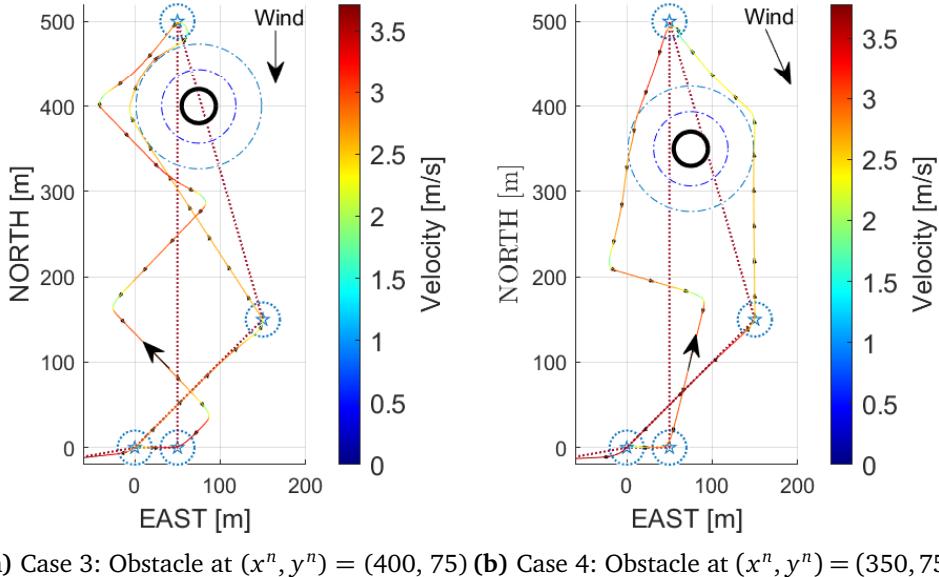


Figure 8.12: Collision avoidance while crossing at 10 m/s wind

(a) Case 3: Obstacle at $(x^n, y^n) = (400, 75)$ (b) Case 4: Obstacle at $(x^n, y^n) = (350, 75)$ **Figure 8.13:** Collision avoidance while crossing at 10 m/s wind

In all simulation cases, the saildrone successfully avoid the obstacle both when using LOS guidance and during crossing. The desired path angles given by the collision avoidance algorithm for the 4 cases are shown in figure 8.14. The actual output of the COLAV module is waypoints, such that both cross-track error and path angle are adjusted. However, it is easier to interpret the path angle alone, and it reflects the behavior of the COLAV module output. From figure 8.14, it can be seen that the path angle is updated every 5 seconds during the avoidance maneuver. The path angle changes in a reasonable and smooth manner, which indicates stability. The small adjustment between the iterations is caused by the approximation and the change in the perception of the obstacle.

The algorithm is based on a greedy choice which is maintained until the path to the target is collision-free. This implies that the longest path might be selected as shown in figure 8.12b. However, without prior knowledge about the obstacle, this can not be determined prior to selection. Other aspects, like COLREG (regulation for collision avoidance at sea), can come into play here, but this is out of scope for this thesis.

Updating the waypoints at constant intervals is not required in a static environment. One could use the algorithm to make a single intermediate waypoint, then reevaluate the algorithm when it is reached, making a path around the obstacle consisting of straight-line segments. However, the used approach has extensions to a dynamic environment where the waypoints need to be adjusted to account for changes in ocean current or obstacle velocities.

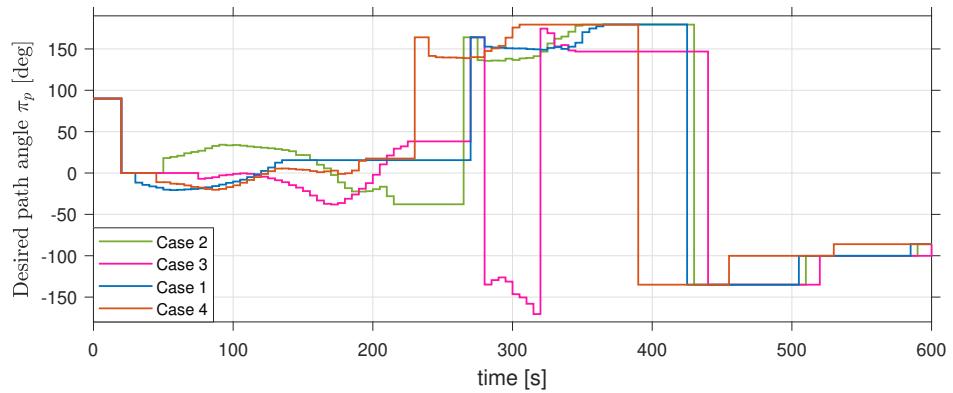


Figure 8.14: Desired path angle during collision avoidance maneuver

Chapter 9

Discussion

The simulation cases studied show that the established motion control system performs well for various environmental conditions. Creating smooth reference signals for the tail and rudder deflection, allowing good reference tracking performance for the heading and sail AOA controller. From the simulations and controller tuning, it can be observed that the rudder often saturates. Meanwhile, this does not introduce any unstable behaviors. However, for tacking maneuvers under harsh conditions, it can be observed that the rudder stalls as a result of the large deflection angle together with the induced roll motion caused by the rudder, keel and sail. This clearly affects the transient heading response, but no stability issues were observed. As the naturally stable roll dynamics is not subjected to control in this thesis, a potential improvement lies in a roll controller. Note that the system has only two inputs, and controllers will need to collaborate. Alternatively, a ballast system could be added as a third input.

Although the simulations show that the motion control system works well within the simulation framework, it should be noted that the simulation is simplified compared to the physical world. Even though the simulation model is highly nonlinear, some effects are left out, like turbulent flows, wind gusts, e.t.c. In addition, the system parameters are based on reasoning and assumptions and might be less realistic. However, it is assumed that the model is representative and captures the dominating dynamics of a saildrone. These are some of the reasons why the utilized drag mode for the sail, might be hard to control in a real-world application. Another simplification is that the control and guidance system run in the simulation loop, where the calculations are always available at the right time. The applied motion control system contains rather simple algorithms, while the saildrone dynamics is relatively slow. Hence computation time has not been the main focus of this study.

The developed gain-scheduled controllers are proven through simulations. Where the shape of the transient response of the system remains similar while it "stretches"

over a more extended period as the relative velocity (with respect to wind or current) decreases. These controllers change bandwidth, and extra care must be taken to preserve sufficient bandwidth separation. To fully exploit the dynamics of the system, the reference models also need to change natural frequency. Despite the extra concerns and implementation, the controllers represent a simple way to use linear theory to derive a controller for this highly nonlinear system.

The sail AOA is always selected to provide maximum forward propulsion in this thesis, assuming it always is desired to keep maximum velocity. This helps maintain controllability and provides robustness for maneuvers. It is also reasonable if it is desirable to do station keeping, as the saildrone has to provide forward velocity also under station keeping, sailing in an 8-shape to maintain controllability. There are, however, cases with extreme wind where it might be desirable to reduce the AOA to avoid capsizing.

The comparison of the course altering and LOS-based approach to upwind tacking shows the conceptual difference between the two methods. In addition, to the potential difference in performance and resulting safety margins towards the dead zone. The simulations indicate that the course approach outperforms the LOS-based one under the tacking maneuver. However, implementing circular-path following for the tacking maneuvers would significantly reduce the difference. Despite this, the comparison is still relevant as LOS guidance and course control have two different objectives. LOS allows deviation from the optimal course to reduce the cross-track error.

The introduction of the ocean current into CVD shows significant improvement and indicates increased robustness under specific environmental conditions. The conducted tests are realistic in terms of parameters as established earlier, particularly along the earth's equator. Another example is when going up narrow passages, where the current can be strong and it is critical not to lose control. The findings are most relevant for low to mid-range velocity crafts, as the improvement decreases as craft velocity increases. This speed range is often used in surveillance and sea monitoring applications. Hence the proposed method can provide increased capacity and robustness under these operations. There is a downside to the method, as it requires an estimate/measurement of the ocean current; additional sensors will be required compared to a method not accounting for the current.

As indicated by the collision avoidance simulation cases, the algorithm enables satisfactory avoidance even when crossing is required. However, under simulation with high sampling time, it was observed that the algorithm tends to give conservative waypoints, as pointed out in the derivation of the approximation used. This is particularly evident when close to small obstacles, as the constraints given by the VO differ a lot between the iterations. Therefore, reducing the sampling time

and increasing the safety margins are remedies to reduce this effect.

Chapter 10

Conclusion

10.1 Conclusion

In this thesis, a motion control system for a sailboat drone is developed and tested for the purpose of collision avoidance and autonomous path following under various environmental conditions. In addition, a framework for simulation and testing has been established, consisting of a highly nonlinear simulation model and a simple perception model.

A simulation model is established, combining Fossen's robot-inspired marine craft model and the open chain dynamics of a robot manipulator. The model represents the saildrone as a chain of rigid bodies, where external forces can be defined within each body. This is convenient when the foil-induced forces acting on the different bodies (keel, rudder, sail and tail) are introduced into the system equations. The resulting model can be represented in matrix form, which is convenient for implementation purposes. A simplified model for control and estimation purposes is also established, introducing the generalized forces of the keel and rudder into Fossen's robot-inspired marine craft model. Through simulation, it is verified that the simple model reflects the dominating dynamics of the established simulation model.

The stability and response of the developed gain-scheduled controllers, together with their respective reference models, are verified through simulation. The shape of the transient response remains similar across the span of velocities. The structure of the controllers represents a simple way to use linear theory to derive a controller for this highly nonlinear system.

To provide maximum forward propulsion, an offline optimization for the sail AOA is derived using the sail lift and drag coefficient models. The resulting lookup table provides the optimal sail AOA as a function of the heading and relative

wind direction. Hence the sail AOA can be selected independent of the path following problem, reducing the complexity of the path following. The expression optimized provides the force components in the desired reference frame, hence it is also used to estimate the sway force in order to cancel these with the keel AOA.

A novel method for incorporating the ocean current into the CVD is derived and utilized to provide the optimal courses when crossing. Simulation shows that the method can significantly increase VMG velocity when crossing under conditions with ocean current. This is most relevant for low and mid-range speeds, as the effect decrease as the saildrone velocity increases. From the simulation cases and the derivation of the method, it is observed that the method provides additional robustness as it accounts for the change of the feasible velocities.

The proposed path following algorithm has been tested through several different simulation cases using the established framework. The simulations show promising results; under all scenarios studied, the algorithm enables the saildrone to follow the desired path even under harsh environmental conditions. Furthermore, the upwind jibing maneuver allows the saildrone to travel upwind even when the obtainable velocity is low. While the increased robustness given by the novel method described above helps provide optimal crossing courses.

The collision avoidance algorithm proposed is tested in simulations and provides satisfactory avoidance in static environments (no current or moving obstacles). The method is rather elementary, however it can be extended to dynamic environments and utilized as a part of more complex avoidance algorithms.

All these contributions are building blocks in a motion control system that can be utilized on an autonomous sailboat. However, each individual module represents features that can be incorporated to reduce the workload on a sailor; automatic heading control, assistance in course selection and crossing, automatic adjustment of the sail, e.t.c. Alternatively, it can be used to maintain the optimal sail angle on a large ship that utilizes sails as secondary propulsion.

10.2 Further work

The motion control system developed through this thesis covers a variety of topics. Hence there are a lot of potential improvements, both with respect to further improvement of the existing subsystems and implementation of others.

To reduce the steps from simulation to real-world application, measurement noise and state estimation for feedback should be implemented to reflect a psychical system better. In addition, the simulations become more realistic if variations in wind and ocean current were included. Another step would be to use hydrodynamic

data from a real saildrone in the simulation model.

As pointed out in chapter 9, a roll controller would be beneficial to reduce the roll rate, potentially improving the turning maneuvers by reducing the stalling of the rudder. An optimization-based controller using multiple states and constraints would also be an interesting solution, as perfecting the turns will yield improved VMG velocity and response.

Another interesting extension of the motion control system would be to design and implement a route optimization layer. Which can find the optimal path from start to finish, utilizing the system constraints, meteorologic data, and maps.

Extending the collision avoidance algorithm to dynamic environments would also be of interest, enabling avoidance of other crafts. Compliance with COLREGs and other perspectives will also come into play here. Extending the method with an optimization would also be an option to improve path decision-making compared to a greedy choice, for example, by accounting for the dynamics. In addition, the utilized perception model represents prepossessed data; hence, a sensing module would also have to be developed for real-world applications or more realistic simulations.

Real-world implementation and testing would also be interesting to verify the methods and performance gain.

Bibliography

- [1] J. Kimball, *Physics of Sailing*. Taylor Francis Inc, Dec. 2009, 296 pp., ISBN: 1420073761. [Online]. Available: https://www.ebook.de/de/product/9341655/john_kimball_physics_of_sailing.html.
- [2] M. F. Silva, A. Fribe, B. Malheiro, P. Guedes, P. Ferreira and M. Waller, ‘Rigid wing sailboats: A state of the art survey,’ *Ocean Engineering*, vol. 187, p. 106150, Sep. 2019. DOI: [10.1016/j.oceaneng.2019.106150](https://doi.org/10.1016/j.oceaneng.2019.106150).
- [3] Q. Sun, Z. Qiao, C. Strömbeck, Y. Qu, H. Liu and H. Qian, ‘Tacking control of an autonomous sailboat based on force polar diagram,’ in *2018 13th World Congress on Intelligent Control and Automation (WCICA)*, 2018, pp. 467–473. DOI: [10.1109/WCICA.2018.8630752](https://doi.org/10.1109/WCICA.2018.8630752).
- [4] L. Xiao, J. C. Alves, N. A. Cruz and J. Jouffroy, ‘Online speed optimization for sailing yachts using extremum seeking,’ in *2012 Oceans*, IEEE, Oct. 2012. DOI: [10.1109/oceans.2012.6404876](https://doi.org/10.1109/oceans.2012.6404876).
- [5] E. Yeh and J.-C. Bin, ‘Fuzzy control for self-steering of a sailboat,’ in *Singapore International Conference on Intelligent Control and Instrumentation [Proceedings 1992]*, IEEE. DOI: [10.1109/sicici.1992.637738](https://doi.org/10.1109/sicici.1992.637738).
- [6] H. Saoud, M.-D. Hua, F. Plumet and F. B. Amar, ‘Optimal sail angle computation for an autonomous sailboat robot,’ in *2015 54th IEEE Conference on Decision and Control (CDC)*, IEEE, Dec. 2015. DOI: [10.1109/cdc.2015.7402329](https://doi.org/10.1109/cdc.2015.7402329).
- [7] L. Xiao and J. Jouffroy, ‘Modeling and nonlinear heading control of sailing yachts,’ *IEEE Journal of Oceanic Engineering*, vol. 39, no. 2, pp. 256–268, Apr. 2014. DOI: [10.1109/joe.2013.2247276](https://doi.org/10.1109/joe.2013.2247276).
- [8] L. Xiao, T. I. Fossen and J. Jouffroy, ‘Nonlinear robust heading control for sailing yachts,’ *IFAC Proceedings Volumes*, vol. 45, no. 27, pp. 404–409, 2012. DOI: [10.3182/20120919-3-it-2046.00069](https://doi.org/10.3182/20120919-3-it-2046.00069).
- [9] R. Stelzer, T. Proll and R. I. John, ‘Fuzzy logic control system for autonomous sailboats,’ in *2007 IEEE International Fuzzy Systems Conference*, IEEE, Jun. 2007. DOI: [10.1109/fuzzy.2007.4295347](https://doi.org/10.1109/fuzzy.2007.4295347).

- [10] D. H. dos Santos and L. M. G. Goncalves, ‘A gain-scheduling control strategy and short-term path optimization with genetic algorithm for autonomous navigation of a sailboat robot,’ *International Journal of Advanced Robotic Systems*, vol. 16, no. 1, p. 172988141882183, Jan. 2019. DOI: 10.1177/1729881418821830.
- [11] L. Jaulin and F. L. Bars, ‘A simple controller for line following of sailboats,’ in *Robotic Sailing 2012*, Springer Berlin Heidelberg, 2013, pp. 117–129. DOI: 10.1007/978-3-642-33084-1_11.
- [12] Y. Dong, N. Wu, J. Qi, X. Chen and C. Hua, ‘Predictive course control and guidance of autonomous unmanned sailboat based on efficient sampled gaussian process,’ *Journal of Marine Science and Engineering*, vol. 9, no. 12, p. 1420, Dec. 2021. DOI: 10.3390/jmse9121420.
- [13] R. Stelzer, *Autonomous Sailboat Navigation*. LAP LAMBERT Academic Publishing, Jun. 2013, 200 pp., ISBN: 3659410500. [Online]. Available: https://www.ebook.de/de/product/20802875/roland_stelzer_autonomous_sailboat_navigation.html.
- [14] C. Petres, M.-A. Romero-Ramirez and F. Plumet, ‘Reactive path planning for autonomous sailboat,’ in *2011 15th International Conference on Advanced Robotics (ICAR)*, IEEE, Jun. 2011. DOI: 10.1109/icar.2011.6088585.
- [15] Z. Cheng, W. Qi, Q. Sun, H. Liu, N. Ding, Z. Sun, T. L. Lam and H. Qian, ‘Obstacle avoidance for autonomous sailboats via reinforcement learning with coarse-to-fine strategy,’ in *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2019, pp. 59–64. DOI: 10.1109/ROBI049542.2019.8961749.
- [16] T. I. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control*, 2nd. WILEY, Apr. 2021, 736 pp., ISBN: 978-1-119-57505-4. [Online]. Available: https://www.ebook.de/de/product/39677368/thor_i_fossen_handbook_of_marine_craft_hydrodynamics_and_motion_control.html.
- [17] O. Egeland, *Modeling and simulation for automatic control*. Trondheim, Norway: Marine Cybernetics, 2002, ISBN: 82-92356-01-0.
- [18] T. W. M. Randal W. Beard, *Small Unmanned Aircraft*. Princeton University Press, Feb. 2012, 320 pp., ISBN: 978-0-691-14921-9. [Online]. Available: https://www.ebook.de/de/product/15359869/randal_w_beard_timothy_w_mcclain_small_unmanned_aircraft.html.
- [19] R. Featherstone, *Robot Dynamics Algorithms*. Springer-Verlag GmbH, Oct. 2007, 211 pp., ISBN: 9780387743158. [Online]. Available: https://www.ebook.de/de/product/11431706/roy_featherstone_robot_dynamics_algorithms.html.
- [20] B. Siciliano, *Springer handbook of robotics*. Berlin: Springer, 2008, ISBN: 9783540239574.

- [21] F. C. P Kevin M. Lynch, *Modern Robotics: Mechanics, Planning, and Control*. CAMBRIDGE, May 2017, 544 pp., ISBN: 1107156300. [Online]. Available: https://www.ebook.de/de/product/29172931/kevin_m_lynch_frank_c_park_modern_robotics_mechanics_planning_and_control.html.
- [22] R. E. Sheldahl and P. C. Klimas, ‘Aerodynamic characteristics of seven symmetrical airfoil sections through 180-degree angle of attack for use in aerodynamic analysis of vertical axis wind turbines,’ Tech. Rep., Mar. 1981. DOI: 10.2172/6548367.
- [23] S. Gudmundsson, *The Anatomy of the Airfoil*. Dec. 2014, pp. 235–297, ISBN: 9780123973085. DOI: 10.1016/B978-0-12-397308-5.00008-8.
- [24] M. V. Cook, *Flight Dynamics Principles Chapter 2 - Systems of Axes and Notation*, Third Edition, M. V. Cook, Ed. Butterworth-Heinemann, 2013, pp. 13–32, ISBN: 978-0-08-098242-7. DOI: <https://doi.org/10.1016/B978-0-08-098242-7.00002-X>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/B978008098242700002X>.
- [25] D. E. Bossert, S. L. Morris, W. F. Hallgren and T. R. Yechout, *Introduction to Aircraft Flight Mechanics*. American Institute of Aeronautics and Astronautics, Jan. 2003. DOI: 10.2514/4.862069.
- [26] A. M. Lekkas and T. I. Fossen, ‘A time-varying lookahead distance guidance law for path following,’ *IFAC Proceedings Volumes*, vol. 45, no. 27, pp. 398–403, 2012, 9th IFAC Conference on Manoeuvring and Control of Marine Craft, ISSN: 1474-6670. DOI: <https://doi.org/10.3182/20120919-3-IT-2046.00068>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1474667016312629>.
- [27] P. Fiorini and Z. Shiller, ‘Motion planning in dynamic environments using velocity obstacles,’ *The International Journal of Robotics Research*, vol. 17, pp. 760–, Jul. 1998. DOI: 10.1177/027836499801700706.
- [28] B. H. Eriksen, M. Breivik, E. F. Wilthil, A. L. Flåten and E. F. Brekke, ‘The branching-course model predictive control algorithm for maritime collision avoidance,’ *Journal of Field Robotics*, vol. 36, no. 7, pp. 1222–1249, Aug. 2019. DOI: 10.1002/rob.21900.
- [29] D. Zu, J. Han and D. Tan, ‘Milp-based trajectory generation in relative velocity coordinates,’ in *2007 46th IEEE Conference on Decision and Control*, 2007, pp. 1975–1980. DOI: 10.1109/CDC.2007.4434566.
- [30] E. V. Lewis, *Principles of Naval Architecture 2nd ed*. The Society of Naval Architects and Marine Engineers (SNAME), 1989, ISBN: 0-939773-01-5.
- [31] O. M. Faltinsen, *Sea Loads on Ships and Offshore Structures*. Cambridge University Press, Sep. 1993, 340 pp., ISBN: 0521458706. [Online]. Available: https://www.ebook.de/de/product/1856644/odd_m_faltinsen_sea_loads_on_ships_and_offshore_structures.html.

- [32] L. Xiao and J. Jouffroy, ‘On motion planning for point-to-point maneuvers for a class of sailing vehicles,’ *Journal of Control Science and Engineering*, vol. 2011, pp. 1–13, 2011. DOI: [10.1155/2011/402017](https://doi.org/10.1155/2011/402017).
- [33] H. Söding, ‘Prediction of ship steering capabilities.,’ vol. 29, pp. 3–29, Mar. 1982.
- [34] D. Xu, J. Cui, R. Bansal, X. Hao, J. Liu, W. Chen and B. S. Peterson, ‘The ellipsoidal area ratio: An alternative anisotropy index for diffusion tensor imaging,’ *Magnetic Resonance Imaging*, vol. 27, no. 3, pp. 311–323, 2009, ISSN: 0730-725X. DOI: <https://doi.org/10.1016/j.mri.2008.07.018>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0730725X08002634>.
- [35] M. S. Klamkin, ‘Elementary approximations to the area of n-dimensional ellipsoids,’ *The American Mathematical Monthly*, vol. 78, no. 3, pp. 280–283, 1971, ISSN: 00029890, 19300972. [Online]. Available: <http://www.jstor.org/stable/2317530> (visited on 28/04/2022).
- [36] D. Spera, ‘Models of lift and drag coefficients of stalled and unstalled airfoils in wind turbines and wind tunnels,’ Jan. 2008. [Online]. Available: <https://ntrs.nasa.gov/citations/20090001311>.
- [37] I. H. Abbott and A. E. Von Doenhoff, *Theory of wing sections, including a summary of airfoil data*, eng, by Ira H. Abbott and Albert E. von Doenhoff, New York, 1959.
- [38] C. Ostowari and D. Naik, ‘Post-stall wind tunnel data for naca 44xx series airfoil sections,’ Jan. 1985. DOI: [10.2172/5791328](https://doi.org/10.2172/5791328). [Online]. Available: <https://www.osti.gov/biblio/5791328>.
- [39] H. Saoud, M.-D. Hua, F. Plumet and F. B. Amar, ‘Routing and course control of an autonomous sailboat,’ in *2015 European Conference on Mobile Robots (ECMR)*, IEEE, Sep. 2015. DOI: [10.1109/ecmr.2015.7324218](https://doi.org/10.1109/ecmr.2015.7324218).
- [40] S. Corbetta, I. Boniolo, S. M. Savaresi, S. Vischi, A. Strassera and D. Manganaris, ‘Real-time identification of the best performances of a sailboat,’ in *Proceedings of the 2011 American Control Conference*, IEEE, Jun. 2011. DOI: [10.1109/acc.2011.5991322](https://doi.org/10.1109/acc.2011.5991322).
- [41] P. Herrero, L. Jaulin, J. Vehí and M. Á. Sainz, ‘Inner and outer approximation of the polar diagram of a sailboat,’ 2005.
- [42] R. Steinegger, ‘The symmetry of wings and sails,’ en, 2019. DOI: [10.21256/ZHAW-18571](https://doi.org/10.21256/ZHAW-18571).
- [43] K. L. Wille, ‘Autonomous sailboats,’ M.S. thesis, Norwegian University of Science and Technology, 2016.
- [44] NASA. ‘Oscar ocean currents with velocity.’ (2012), [Online]. Available: <https://svs.gsfc.nasa.gov/3958> (visited on 30/04/2022).
- [45] P. Webb, *Introduction to oceanography*. Roger Williams University, 2019.

- [46] J. L. Meriam, *Dynamics*. Hoboken, NJ: Wiley, 2012, ISBN: 978-1-118-08345-1.

Appendix **A**

Path following algorithm pseudo-code

Remark; the algorithm is generic for any type of waypoints; hence both local and global waypoints are described as WP in algorithm 3, while the path angle is described by π_p . During normal operation, the algorithm will receive GWP from the path manager, while under collision avoidance, it gets local waypoints.

To describe algorithm 3, the parameters from figure 7.8 together with the states and variables in table A.1 are used.

Table A.1: State and variable definitions algorithm 3

Description	Notation
The crossing type state, which is 1 if the saildrone goes to the right and 0 if it is going to the left from the perceptive of the path coordinate frame $\{p\}$ seen in figure 7.8.	$typeC$
State that indicate if crossing is required.	X
State that determine if upwind jibing is enabled.	ssa_{on}
Threshold angle for when the WP^- can be updated.	t_{LOS}
The width of the upwind jib initialization region shown in yellow in figure 7.8.	jib_{margin}
The deviation between the actual and desired course at with a upwind jibing maneuver is terminated.	$reset_{limit}$

The notion X^- and χ_d^- are the previous crossing state and desired course.

Algorithm 3 Path following algorithm

Input: $WP^+, WP^-, x^n, y^n, \chi, U_{rc}, \beta_c, U_c, \beta_w, U_w$
Output: χ_d, ssa_{on}
Initialize: $\chi_d^-, \Delta, X^-, ssa_{on}, t_{LOS}, d_L, d_{Lj}, d_R, d_{Rj}, jib_{margin}, jib_{angle}, reset_{limit}, U_{rc_{limit}}, typeC$

```

 $\pi_p \leftarrow \text{atan2}(WP^+(2)-WP^-(2), WP^+(1)-WP^-(1))$ 
 $\chi_t \leftarrow \text{atan2}(WP^+(2)-y^n, WP^+(1)-x^n)$ 
 $y_e^p \leftarrow \text{Compute cross track error from } \pi_p \text{ using equation 2.41}$ 
 $[\chi_{x_R}, \chi_{x_L}, X] \leftarrow \text{Compute crossing angles and if crossing is needed given the path direction } \pi_p \text{ using section 7.1.}$ 

%% Rotate courses into the path frame
 $\chi_{x_R}^p \leftarrow ssa(\chi_{x_R} - \pi_p)$ 
 $\chi_{x_L}^p \leftarrow ssa(\chi_{x_L} - \pi_p)$ 
 $\chi_t^p \leftarrow ssa(\chi_t - \pi_p)$ 

reused  $\leftarrow 0$ 

%% Select the desired course
if ( $X == 1$  and  $X^- == 1$ ) then
  if ( $\chi_t^p > \chi_{x_R}^p + t_{LOS}$  or  $\chi_t^p < \chi_{x_L}^p - t_{LOS}$ ) then
    if  $ssa_{on} == 1$  then
       $WP^- \leftarrow [x^n, y^n]$ 
       $X \leftarrow 0$ 
    end if
  end if
end if

if  $typeC == 1$  then
  if  $y_e^p < d_R$  then
     $\chi_d \leftarrow \chi_{x_R}^p$                                 ▷ Going right
  else
     $\chi_d \leftarrow \chi_{x_L}^p$                             ▷ Turn around
     $typeC \leftarrow 0$                                     ▷ Not crossed limit yet
  end if
else
  if  $y_e^p > d_L$  then
     $\chi_d \leftarrow \chi_{x_L}^p$                             ▷ Keep going left
  else
     $\chi_d \leftarrow \chi_{x_R}^p$                             ▷ Turn around
     $typeC \leftarrow 1$                                     ▷ Not crossed limit yet
  end if
end if

```

```

else if ( $X == 1$  and  $X^- == 0$ ) then  $\triangleright$  Transition and selection of first direction
  if  $abs(y_e^p) < min(-d_{Lj}, d_{Rj})$  then  $\triangleright$  Inside the inner path constraints
     $typeC \in \{0, 1\}$  Select the crossing course closest to the current
  else if  $y_e^p >= d_R$  then  $\triangleright$  To far to the right
     $typeC \leftarrow 0$ 
  else  $\triangleright$  To far to the left
     $typeC \leftarrow 1$ 
  end if
   $chi_d \leftarrow chi_d^-$ 
   $reused \leftarrow 1$ 
else
  if  $ssa_{on} == 1$  then  $\triangleright$  Use LOS
     $K_p \leftarrow 1/\Delta$ 
     $\chi_d \leftarrow \pi_p - atan(K_p y_e^p)$ 
  else
     $\chi_d \leftarrow \chi_t$ 
  end if
end if

%% When to jib upwind
if  $U_{rc} < U_{rc_{limit}}$  then  $\triangleright$  The relative velocity is to low to tack
  if  $abs(ssa(\pi_p - \beta_w + \pi)) < pi/2$  then  $\triangleright$  Path goes towards the wind
    if ( $X == 1$  and  $X^- == 1$ ) then
      if  $d_{Rj} - jib_{margin} < y_e^p < d_{Rj}$  then
        if  $typeC == 1$  then
           $ssa_{on} \leftarrow 0$ 
           $\chi_d \leftarrow \chi_{X_L}$ 
           $typeC \leftarrow 0$ 
        end if
      else if  $d_{Lj} + jib_{margin} > y_e^p > d_{Lj}$  then
        if  $typeC == 0$  then
           $ssa_{on} \leftarrow 0$ 
           $\chi \leftarrow \chi_{X_R}$ 
           $typeC \leftarrow 1$ 
        end if
      end if
    end if
  end if
end if
if  $abs(ssa(\chi - \beta_w + \pi)) < pi/2$  then  $\triangleright$  Current direction is into the wind
  if  $ssa(\chi - \beta_w + \pi) > 0$  and  $ssa(\chi_d - \beta_w + \pi) < 0$  then  $\triangleright$  Cross the wind
    if  $ssa(\chi_d - \chi) < 0$  then
       $ssa_{on} \leftarrow 0$ 
    end if

```

```

else if  $ssa(\chi - \beta_w + \pi) < 0$  and  $ssa(\chi_d - \beta_w + \pi) > 0$  then
    if  $ssa(\chi_d - \chi) > 0$  then
         $ssa_{on} \leftarrow 0$ 
    end if
    end if
end if

%% Use longest angle to jib upwind
if  $ssa_{on} == 0$  then
    if  $jib_{angle} == nan$  and  $reused == 0$  then
         $jib_{angle} \in \{0, 2\pi\}$  Select the angle that give the longest turn
    end if
     $\chi_d \leftarrow \chi_d - jib_{angle}$ 
end if

%% Upwind jib deactivation when error has been reduced
if  $abs(ssa(\chi_d - \chi)) < reset_{limit}$  then
     $ssa_{on} \leftarrow 1$ 
     $jib_{angle} \leftarrow nan$ 
end if

 $X^- \leftarrow X$ 
 $\chi_d^- \leftarrow \chi_d$ 

```

Aerodynamic data of airfoil

Wind tunnel data for a set of NACA (National Advisory Committee for Aeronautics) standard foils are presented in reference [22], where tests are conducted for angle of attack in the range 0-180 degrees. Here the applied data from the NACA-0015 wing is presented. The data is represented at the quarter chord.

The data show the section characteristic coefficients, which represent the coefficients given a 2D cross section of a foil, by utilizing the concept of infinite aspect ratio [37]. Which enable the study of foils regardless of there aspect ratio.

96

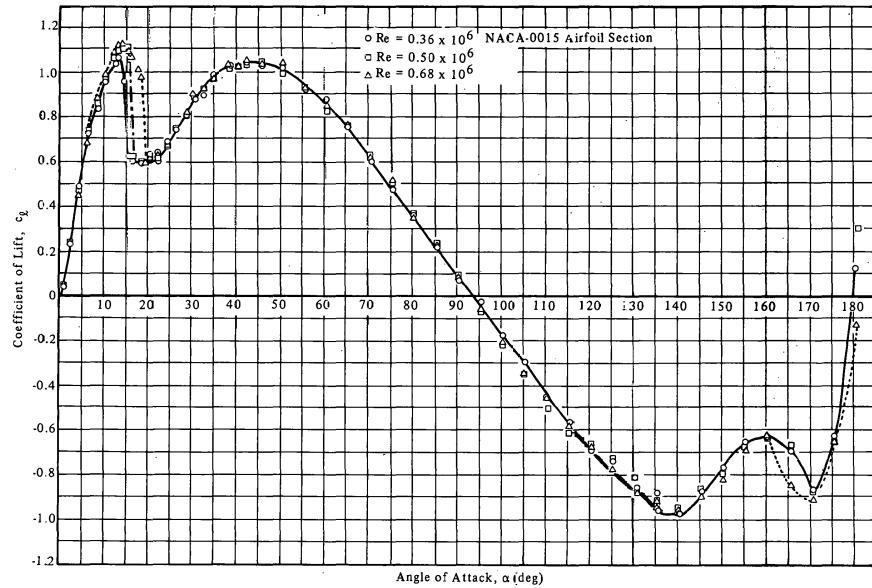


Figure B.1: NACA-0015 experimental lift coefficient

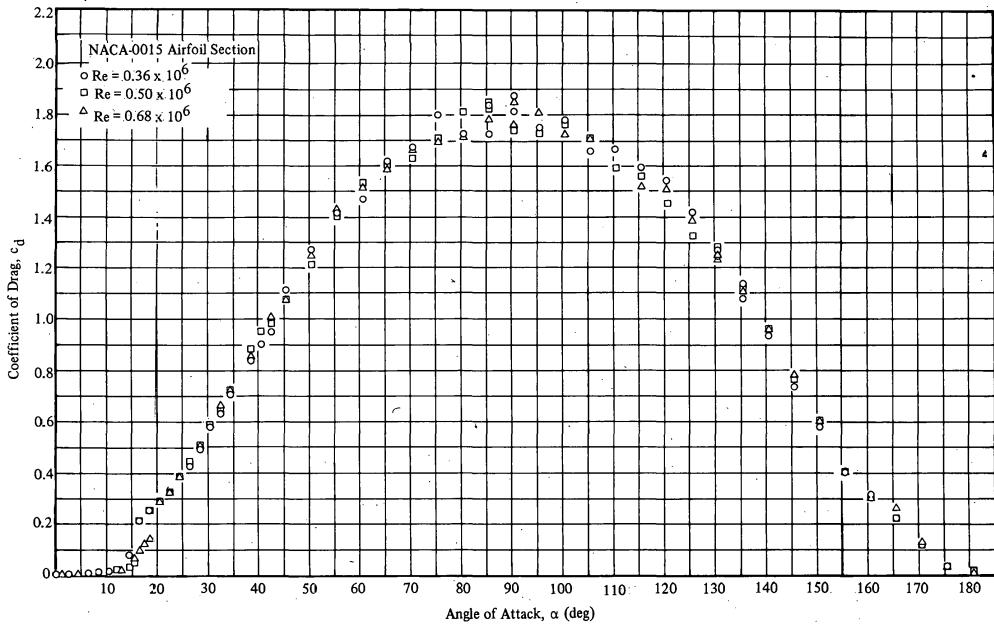


Figure B.2: NACA-0015 experimental drag coefficient

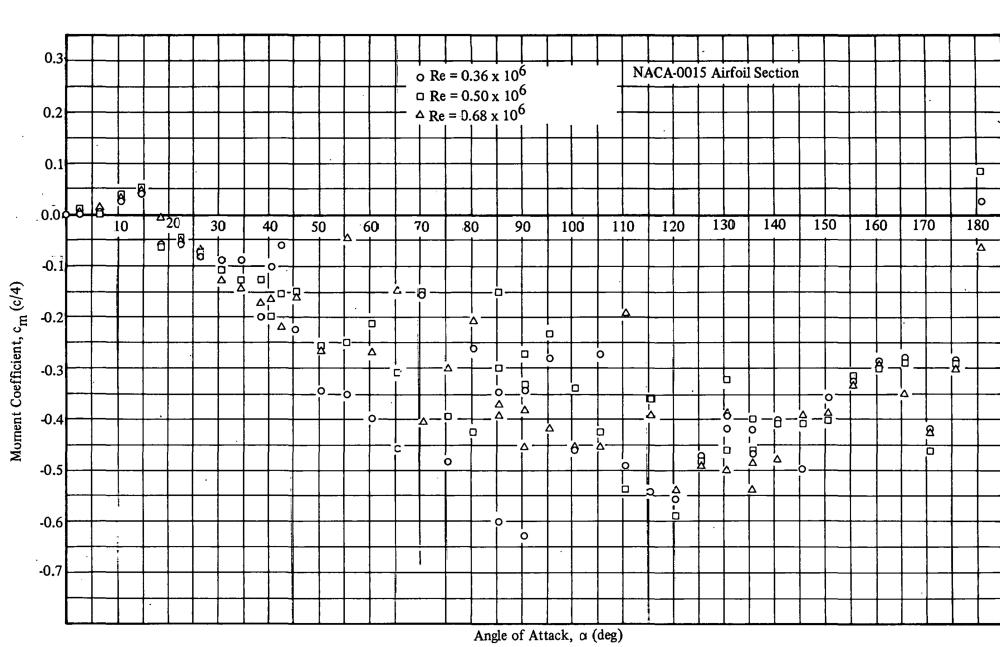


Figure B.3: NACA-0015 experimental moment coefficient

Properties of homogeneous solids

The formulas in this section are gathered from reference [46] in order to calculate moment of inertia of rigid bodies. Along with parallel axis theorem taken from [16]. The inertia in the formulas are given with respect to the center of gravity of the rigid body.

Inertia of a elliptical cylinder:

$$I_{xx} = \frac{1}{4}ma^2 + \frac{1}{12}ml^2 \quad (\text{C.1a})$$

$$I_{yy} = \frac{1}{4}mb^2 + \frac{1}{12}ml^2 \quad (\text{C.1b})$$

$$I_{zz} = \frac{1}{4}m(a^2 + b^2) \quad (\text{C.1c})$$

Where a and b are the semi axes length, while l is the length of the cylinder.

Inertia of a rectangular parallelepiped:

$$I_{xx} = \frac{1}{12}m(a^2 + l^2) \quad (\text{C.2a})$$

$$I_{yy} = \frac{1}{12}m(b^2 + l^2) \quad (\text{C.2b})$$

$$I_{zz} = \frac{1}{12}m(a^2 + b^2) \quad (\text{C.2c})$$

Where a , b and l are the length of the rectangle sides.

Inertia of a ellipsoid:

$$I_{xx} = \frac{1}{5}m(b^2 + c^2) \quad (\text{C.3a})$$

$$I_{yy} = \frac{1}{5}m(a^2 + c^2) \quad (\text{C.3b})$$

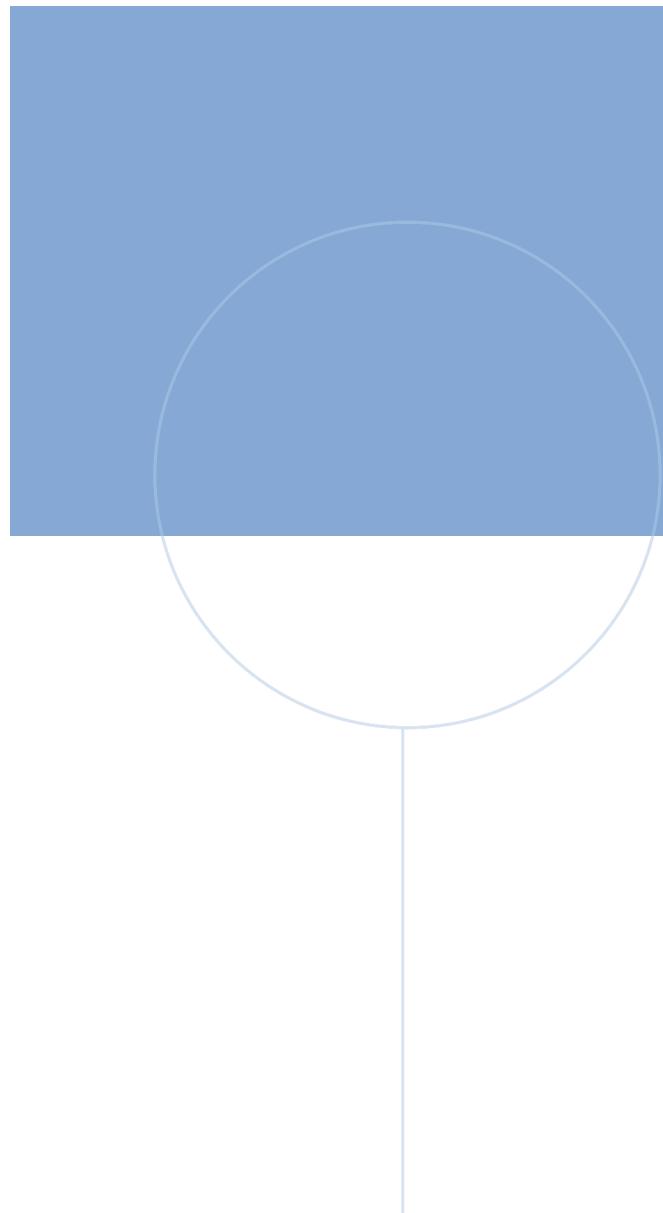
$$I_{zz} = \frac{1}{5}m(a^2 + b^2) \quad (\text{C.3c})$$

Where a , b and c are the length of the 3 semi axes.

Parallel-axis theorem:

$$I_b^b = I_g^b - mS^2(r_{bg}^b) \quad (\text{C.4})$$

Where I_b^b is the inertia at frame b 's origin, while I_g^b is the inertia at CG with respect to the axis of frame b . r_{bg}^b is the vector from b 's origin to CG with respect to b 's axis. This relation holds for transformation between point within the same rigid body.



NTNU

Norwegian University of
Science and Technology