Robust Relative Positioning & Autonomous Landing of a Flying Wing MAV for Sea Rescue Applications

Olle Fagerström Hedbrant

Martin Gemborn Nilsson

LUND UNIVERSITY

Department of Automatic Control

MSc Thesis
TFRT-6104
ISSN 0280-5316

Department of Automatic Control
Lund University
Box 118
SE-221 00 LUND
Sweden

Printed in Sweden by Tryckeriet i E-huset
Lund 2020

Abstract

In this thesis, a study is made into relative positioning and control systems for flying wing Micro Aerial Vehicles (MAVs) intended to be used by the Swedish Sea Rescue Society (SSRS) for the purpose of providing assistance during sea rescue missions. The particular scenario of interest is robust relative positioning to be used for autonomous landing of a MAV on a boat at the end of a rescue mission. Three positioning methods, Global Navigation Satellite System (GNSS), 3D LiDAR and a scaled down version of an Instrument Landing System (ILS) are explored and evaluated through physical testing, modeling and simulation. In addition, a control system based on Model Predictive Control (MPC) is proposed for autonomous landing of a flying-wing MAV on a moving boat. A full simulation of the target scenario, including positioning systems, control system and environmental factors is made, from which conclusions about the feasibility of successfully performing a landing for the combined positioning system and landing algorithm are drawn. The results of the thesis indicate, given the proposed landing strategy, that a high performance GNSS would be sufficient to complete this type of landing consistently. The 3D LiDAR solution also shows potential, but is based on a number of assumptions and simplifications in modeling. The small scale ILS method proposed suffered from precision issues that make it unfit for the target scenario.

Acknowledgements

We would like to thank our academic supervisors Anders Robertsson and Marcus Greiff for all their efforts and dedication in helping us throughout the project. Your commitment to education and continuous encouragement for us to always learn new things have been a great contributor of motivation during the project. We would also like to thank Anders J. Johansson for helping us with radio communication and antennas. Thank you also to Fredrik Falkman at the Swedish Sea Rescue Society for letting us be a part of the exciting journey of figuring out how high tech autonomous systems can be incorporated into the sea rescue world. A final thank you goes to our families and friends, for their support and all unforgettable moments and experiences shared during our time at Lund University.

Contents

1

Introduction

In recent years, the use of Micro Aerial Vehicles (MAVs) has risen substantially with applications spanning from geographical mapping and visual crop surveillance in agriculture to distribution of medicine to remote areas. The small form factor and low weight make it possible to, at a low cost, cover large distances and areas quickly. Lately MAVs has also gained popularity in search and rescue missions, where they serve as an excellent tool for video feedback.

Figure 1.1: One of the rescue boats in the SSRS fleet. [*SSRS rescue fleet* 2020]

At the Swedish Sea Rescue Society (SSRS), a non profit sea rescue organization mainly operated by volunteers, efforts are being made to incorporate autonomous systems and drone technology into their mission routines. Today, if incidents occur at sea, boats are sent out to help and assist. An example of an SSRS rescue boat can be seen in Figure 1.1. By deploying a flying-wing MAV instantly when an

incident is called in, video feedback from the scene of incident could be provided to the rescue-team on its way or even before the departure of the rescue boat(s). This could give access to valuable information for the rescue-team on what to prepare for, the severity of the situation and what equipment to bring. Another scenario where MAVs could be of great value is if the exact site of the incident is unknown. Then a MAV can fly in a search pattern with the goal of more accurately locating the site of incident.

1.1 Background

To maximize the utility of the MAV in a sea rescue mission, as much time as possible should be spent at mission related tasks. When a rescue mission is finished, the MAV somehow has to land. This requires the MAV to either fly back to an appropriate landing spot at the coast, to land in the sea or to land on one of the boats. In the former case, the limited battery capacity of the MAV can severely shorten the time of operation spent at the scene of incident. Because of this, it is preferable to go with one of the latter alternatives and spend all the available energy in a more useful way. Landing in the sea is the simple solution but it also introduces other problems. If the MAV is to be used for multiple missions, loosing or damaging the MAV is highly undesirable. A landing in the sea increases the risk of getting hit by waves and would also require the MAV to be completely water-proof. Landing on a boat is technically more advanced, but does not suffer from any of the disadvantages mentioned above. In order to land the MAV on a boat, there is need for a robust relative positioning system that is good enough to accomplish the task. Using the strategy of landing on a boat, an increase in cost is acceptable since it might allow the MAV to be reused many times. However, adding weight to the MAV should be avoided to the highest extent possible, as extra weight also lowers the maximum flight-time.

There are other applications related to sea rescue operations that might benefit from a robust relative positioning system as well. Examples includes to autonomously control a rescue runner or a tethered drone to follow a boat. It is not certain that a positioning system working well for one of these applications is equally suitable for another application. However, because of the performance needed and the relatively large relative velocities and distances involved, the landing of a flying wing drone on a boat is arguably the most demanding case for such a positioning system. This is why the thesis is focusing on this particular topic.

Aside from a positioning system, a robust control algorithm is required to guide the MAV to its destination. In general, the positioning system and the control system are dependent on each other in several ways. This because different kinds of relative positional information might influence the design of the control strategy, or depend-

ing on the control strategy, there could be different requirements on the positioning system. In light of this, the inter-relationship between the two makes it necessary to study them together.

1.2 Target Scenario

The target scenario described below reflects SSRS' current belief on how rescue operations aided by autonomous MAVs could unfold. The mission starts with an incident alarm being received. From information on where the incident has occurred as well as the current weather conditions, a software finds the MAV with the best accessibility to the scene of incident. An operator authorizes the mission and the MAV is then automatically deployed and flies towards the site of incident. During flight, the MAV is guided by so called way-points, defining straight line segment for the MAV to follow. The way-points are set manually by an operator or found by an algorithm, either making sure to reach the destination as fast as possible while keeping the flight safe, for example by avoiding controlled airspace. Upon arrival the MAV would loiter the scene while recording and sending back a video stream to the operator where it, in real time, is distributed to the relevant people. When the rescue boat(s) has arrived and attended to the incident the landing phase can be initiated.

In order minimize the impact of the MAV upon landing and maximize the chances of a successful landing, the relative velocity of the boat and MAV should be kept to a minimum. To achieve this, the boat could navigate forward with a certain velocity while the MAV approaches from the rear. In most cases the relative velocity is limited by the maximum convenient speed of the boat. Since the type of boat, weather and wave conditions, and surrounding environment can be expected to vary, the maximum velocity of the boat will change as well. Depending on weather and wave conditions, the velocity of the boat is estimated to be between 2 and 8 m/s. If no obstructions exist, the boat will be able to line itself up towards the wind in order to further minimize the relative velocity as this will allow the MAV to fly slower in relation to the ground.

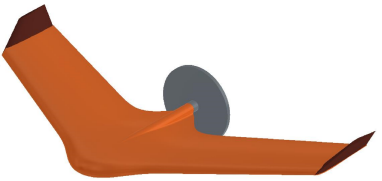
Since the MAV is very lightweight, the actual landing impact can be achieved by letting a crew member, positioned in the rear of the boat facing backwards, catch the MAV with their hands. With this in mind, the MAV target is defined as a 1 m radius circle in the vertical plane, 2 m over the water-surface. This reflects an approximation of the area which a crew member can reach while standing still.

MAV Setup

Multiple MAV setups are currently used by SSRS for development and testing purposes. Most frequently, a Parrot Disco airframe, which can be seen in Figure

1.2a, is used. A custom designed airframe is intended to be used in the future and can be seen in Figure 1.2b.

(a) Parrot Disco airframe used for development and testing.



(b) A custom designed airframe.

Figure 1.2: Flying-wing airframes of importance to SSRS.

Except the airframe, a typical MAV setup currently includes the following components

- A flight computer (including IMU and barometer) running Ardupilot or PX4 software.
- A 4G (LTE) cellular data communication link.
- A Raspberry Pi computer capable of communication with the operator (through the cellular communication link), and the flight computer running PX4 or Ardupilot.
- Motor, Electronic Speed Controllers (ESC) and servos for control of speed and control surfaces.
- Batteries powering the MAV.
- Additional sensors such as pitot tube for airspeed, laser range finder for accurate altitude measurements and GNSS for global positioning.
- Camera(s) used for capturing video to be streamed to the operator.

1.3 Scope

This section states the problem formulation, a brief description what has been done within the thesis, and what delimitations have been made.

Problem Formulation

With the target scenario set, the problem can be formulated. The objectives of this thesis are the following:

- Investigate feasible solutions for robust relative positioning between a MAV and boat according to the target scenario.
- Design and evaluate an appropriate control algorithm that can be used for autonomous landing of a flying-wing MAV on a boat.
- Make a real-world implementation of a combined relative positioning system and landing algorithm.

Extent of Project

For this thesis, three different kinds of positioning systems were explored as alternatives for the target scenario. Analysis of the systems were made regarding how well they would fit the end goal of providing robust positioning. To evaluate the theoretical performance, the positioning systems were modeled and used in full system simulations. Tests were performed on the physically available systems and some of the results and conclusions drawn were incorporated into the full system simulation to make it more realistic.

A control system and corresponding landing strategy were also designed and evaluated. The main focus was on implementing a Model Predictive Controller (MPC) and to make use of its strengths in order to complete a landing strategy.

A complete simulation, including the different positions systems, the boat, the MAV and the proposed control system was implemented in Matlab. The simulation also includes a Dryden wind model and a simple sinusoidal model of waves. Naturally, the parts of the simulations identified as more important have been given more focus than less important parts. The simulation is meant to evaluate the feasibility of a few positioning-control system pairs by looking at flight trajectories, where the MAV believes it hits the boat, and where the MAV actually hits the boat.

The initial idea was to physically implement the most promising positioning system if possible. The plan was to also implement the chosen control-system on a physical MAV of similar size to the ones used by SSRS, and make real world test of the full system. Unfortunately neither of those ideas came nearly as far as initially planned. General time constraints, together with a world wide pandemic (COVID-19), causing extra delays and logistics problems, made it impossible to realize the initial scope within an acceptable time-frame. However, some initial tests, designs and implementations were done, and these are described briefly in the report although not completed. Thus, a few additional experiments and designs that might

be helpful for future implementation were added to the report, even if they do not directly relate to the contents of the results in Chapter 6.

Delimitation

For positioning systems, only a narrow set of alternatives were explored compared to all previously proposed systems in this area of research. Most notably, vision based systems were not considered. Because of the problems rain can cause vision based systems, including rain on the lense and sun glares in the water, this approach to positioning was considered too fragile for further studies for this particular purpose.

On the control system side the focus has been MPC. Other approaches such as PID-controllers could have been explored in more detail but was not considered promising enough to be prioritized among many other areas important to address. Also, some functionality needed for the MAV to land was not considered in this project. Examples include decisions on when to abort a landing, and functionality that is already directly available in the flight stacks used by SSRS. Examples of such available functionalities are the MAV state estimator and estimation of ambient wind. Ideally, for the purpose of simulation, an approximation of the covariances for the outputs of these state estimators should have been identified for the real-world MAV. However this would require a flying MAV as well as a ground truth, neither being available.

Division of Work

Both authors have worked with vehicle modeling and general tasks necessary for the thesis such as project planning, report writing and general administration. Apart from this, Olle has been working with evaluation, testing and simulation of the positioning systems while Martin has been focusing on control system design and simulation of landing scenarios.

1.4 Related Work

Before this thesis, several other masters theses and university projects have been conducted to serve the purpose of including new technology into SSRS search and recuse missions. This includes a launchpad for autonomous flying wing takeoffs [Valero Beltrá, 2018], control systems for rescue-runners to autonomously follow a boat [Voigt and Alkaysi, 2020] and an optimization-based strategy for autonomous landing for flying-wing MAV on land [Fridén, 2020]. In addition a mission specific flying wing airframe has been designed as part of a master's thesis project, which at the time of writing has not been published. References to this airframe are made throughout this thesis and relevant information about it has been compiled in Ap-

pendix C.

Positioning of small unmanned aerial vehicles (UAVs) has been researched in many different ways. For indoor positioning, several methods are presented in [Pérez et al., 2018], some of which include Ultra Wide-Band (UWB), vision based systems, WLAN and Inertial Measurement Unit (IMU). Methods for outdoor purposes traditionally include Global Navigation Satellite System (GNSS) systems and sensor fusion methods using an IMU. Recent research has studied 3D LiDAR devices and object detection, as in [Hammer et al., 2018].

Quite a few studies on autonomous landing of UAVs have been conducted. Many of these studies have been focusing on rotor UAVs aided with computer vision techniques to perform the landing. However, the case of using a fixed wing and a copter drone are quite different and vision based approaches might come short outdoors when fog, mist or glares come into play. A summary of some studies, both for fixed-wing and copter style drones, can be found in [Gautam et al., 2014]. Previous work on landings of fixed-wing UAVs on moving targets, similar to the situation in this thesis, has been done in [Muskardin et al., 2017] and [Persson, 2019]. First, experimentally using a PID-based control approach and then in simulation with a MPC-based approach. In these studies cooperative landing has been used, i.e. when both aircraft and target vehicle are controlled simultaneously by the landing algorithm, which will not be the case for this thesis. The concept of using a MPC to perform autonomous landing with a small fixed-wing MAV has also been explored in [Mathisen et al., 2016], here using a deep stall approach.

1.5 Outline

In Chapter 2 the report starts by presenting notations, definitions and the mathematical models used for the simulation. Chapter 3 explores the positioning side of the problem and three positioning methods are analysed and evaluated. In Chapter 4, a control system and corresponding strategies for landing on a moving target is proposed. After this, Chapter 5 combines Chapter 3 and Chapter 4 into a full simulation, also including environmental factors such as wind and waves. In Chapter 6, the results of the simulation is presented and chapter 7 facilitates a short discussion on steps taken towards a real-world implementation, including simple real-world small scale test on a RC car. Chapter 8 concludes the thesis, discussing and reflecting upon the contents of the report. Suggestions on how to proceed with the project in the future is also included.

2

Modeling

In this chapter, the relevant models used in the simulations are presented. This includes an introduction to the reference frames used as well as mathematical models for the MAV, the boat, and external factors such as wind and waves.

2.1 Reference Frames

To facilitate a discussion of relative movement of mechanical bodies, the concept of reference frames is introduced. This is done in order to derive and keep track of relationships between kinematics and dynamics in a way as easy and intuitive as possible. In situations where large spatial distances are considered, the curvature of earth will have significant impact on relative positions and orientations. However, this will not be the case for this report and thus regular Cartesian coordinates will be used. A few reference frames will be of particular interest; inertial frame ($\mathcal{I}$), vehicle frame ($\mathcal{V}$), vehicle-1 frame ($\mathcal{V}_1$), body frame ($\mathcal{B}$), stability frame ($\mathcal{S}$) and wind frame ($\mathcal{W}$). A rotation of a vector $\mathbf{a}$ from $\mathcal{I}$ to $\mathcal{B}$ will be denoted by

$$\mathbf{a}^{\mathcal{B}} = R_{\mathcal{I}}^{\mathcal{B}} \mathbf{a}^{\mathcal{I}}. \tag{2.1}$$

where $R_{\mathcal{I}}^{\mathcal{B}}$ is a matrix defining the rotation, and the vector $\mathbf{a}$ is expressed in a specific frame by the following notation

$$\mathbf{a}^{\mathcal{I}} = \mathbf{i}_x^{\mathcal{I}} a_x + \mathbf{j}_y^{\mathcal{I}} a_y + \mathbf{k}_z^{\mathcal{I}} a_z = (a_x, a_y, a_z)^T \tag{2.2}$$

where $\mathbf{i}_x$, $\mathbf{j}_y$ and $\mathbf{k}_z$ are basis vectors of that particular frame. The rotations of the frames themselves are defined by positive rotations around the axis of rotation. Thus, the corresponding transformation of vectors between the frames, are negative rotations around the same axis of rotation.

Inertial Frame

The inertial frame ($\mathcal{I}$) used in this report is an earth-fixed, Cartesian coordinate system, with the origin defined at some home location. The principal axis are defined with $\mathbf{i}_x^{\mathcal{I}}$ pointing north, $\mathbf{i}_y^{\mathcal{I}}$ pointing east and $\mathbf{i}_z^{\mathcal{I}}$ pointing down. This convention

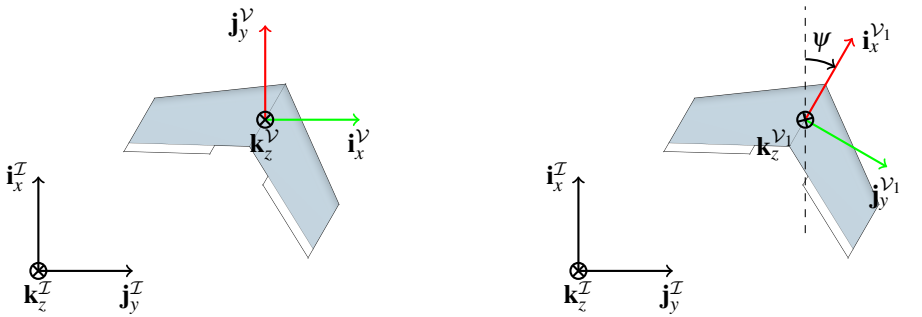
is standard for aircraft and marine applications and is sometimes referred to as a north-east-down (NED) reference frame.

Vehicle Frame

The next frame of importance is the vehicle frame ($\mathcal{V}$). The origin of $\mathcal{V}$ is located in the center of gravity of the vehicle and is using the same directions as the inertial frame for its principal axis (i.e north-east-down respectively), see Figure 2.1a. The transformation from $\mathcal{I}$ to $\mathcal{V}$ is simply a translation along the three axis in the inertial frame.

Vehicle-1 Frame

We also introduce the vehicle-1 frame ($\mathcal{V}_1$). This frame is defined by rotating $\mathcal{V}$ around $\mathbf{k}_z^{\mathcal{V}}$ by an angle ψ (later referred to as yaw) such that $\mathbf{i}_x^{\mathcal{V}_1}$ is pointing in the same horizontal direction as the vehicle (see Figure 2.1b).



(a) Illustration of the relation between $\mathcal{I}$ and $\mathcal{V}$.

(b) Inertial frame ($\mathcal{I}$) and vehicle-1 frame ($\mathcal{V}_1$).

Figure 2.1: Illustration of the relations between frames $\mathcal{I}$, $\mathcal{V}$ and $\mathcal{V}_1$, with the MAV seen from above.

Body Frame

The body frame ($\mathcal{B}$) also has its origin in the vehicle center of gravity, but with its principal axes attached to the body of the vehicle. The principal axes in $\mathcal{B}$ are defined with $\mathbf{i}_x^{\mathcal{B}}$ pointing in the forward direction of the vehicle, $\mathbf{j}_y^{\mathcal{B}}$ pointing through the right side of the vehicle and $\mathbf{k}_z^{\mathcal{B}}$ pointing through the bottom of the vehicle (see Figure 2.2).

The transformation between the vehicle frame and the body frame is a single rotation. This rotation can be described in a more intuitive way by splitting it up in three successive rotations using the *intrinsic ZYX Tait-Bryan* representation. First the coordinate frame is rotated around the $\mathbf{k}_z^{\mathcal{V}}$-axis by angle ψ (resulting in $\mathcal{V}_1$). The

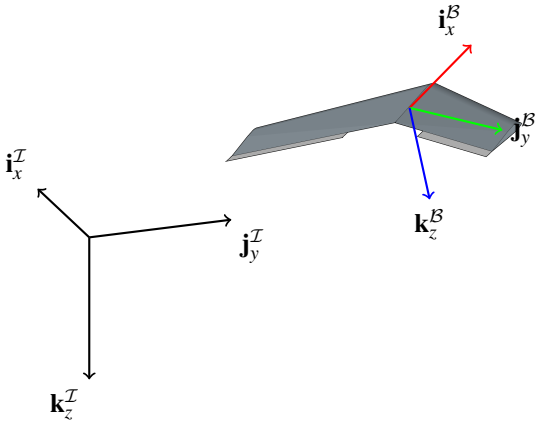


Figure 2.2: Illustration of the relation between inertial frame ($\mathcal{I}$) and body frame ($\mathcal{B}$).

second rotation is around the $\mathbf{j}_y^{\mathcal{V}_1}$-axis by angle θ (later referred to as pitch). The third rotation is around the new $\mathbf{i}_x$-axis by angle ϕ (later referred to as roll). For a vector in $\mathcal{V}$, each one of these rotations can be expressed by a matrix. Multiplying these matrices together gives the final transformation of a vector from the vehicle frame ($\mathcal{V}$) to the body frame ($\mathcal{B}$). The rotation of the *frames* themselves are defined by *positive* rotations. Thus, the corresponding rotations that transform the *vectors* from one frame to another, are *negative* rotations. The rotation-matrix transforming a vector from $\mathcal{V}$ to $\mathcal{B}$ is given by

$$R_{\mathcal{V}}^{\mathcal{B}} = R_x(\phi)R_y(\phi)R_z(\psi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & c_\phi & s_\phi \\ 0 & -s_\phi & c_\phi \end{pmatrix} \begin{pmatrix} c_\theta & 0 & -s_\theta \\ 0 & 1 & 0 \\ s_\theta & 0 & c_\theta \end{pmatrix} \begin{pmatrix} c_\psi & s_\psi & 0 \\ -s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{2.3}$$

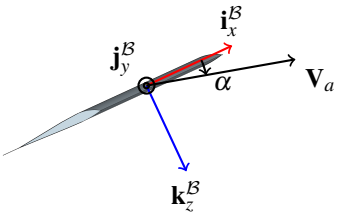
$$R_{\mathcal{V}}^{\mathcal{B}} = \begin{pmatrix} c_\theta c_\psi & c_\theta s_\psi & -s_\theta \\ s_\phi s_\theta c_\psi - c_\phi s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & s_\phi c_\theta \\ c_\phi s_\theta c_\psi + s_\phi s_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi & c_\phi c_\theta \end{pmatrix} \tag{2.4}$$

where $c_\phi \triangleq cos(\phi)$ and $sin(\phi) \triangleq s_\phi$.

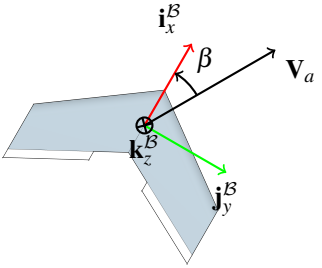
Stability Frame and Wind Frame

The velocity of the MAV in relation to the surrounding air is called the airspeed-vector ($\mathbf{V}_a$) and the magnitude of $\mathbf{V}_a$ is called airspeed (V_a). Angle of attack (α) is defined as the angle from $\mathbf{i}_x^{\mathcal{B}}$ to the projection of $\mathbf{V}_a$ into the $\mathbf{i}_x^{\mathcal{B}}\mathbf{k}_z^{\mathcal{B}}$-plane (see Figure 2.3a). We also define the side-slip angle (β) which is the angle from $\mathbf{V}_a$ to

its projection onto the $\mathbf{i}_x^{\mathcal{B}}\mathbf{k}_z^{\mathcal{B}}$-plane (see Figure 2.3b).



(a) Angle of attack, α from a side view of the flying wing. Notice that this is a simplified case where roll is zero (i.e. $\phi = 0$).



(b) Slip angle β from a top-view of the flying-wing. Notice that this is a simplified case where $\mathcal{B}$ is aligned with $\mathcal{V}_1$ (i.e. $\theta = \phi = 0$).

Figure 2.3: Illustration of angle of attack (α) and side slip angle (β).

The wind frame ($\mathcal{W}$) has its origin in the center of gravity of the MAV and is defined by rotating $\mathcal{B}$ with two intrinsic rotations. First, a negative rotation around $\mathbf{j}_y^{\mathcal{B}}$ by angle α, resulting in an intermediate frame called stability frame ($\mathcal{S}$). Continuing with a positive rotation around $\mathbf{k}_z^{\mathcal{S}}$ by angle β, finally results as $\mathcal{W}$. The rotation taking a vector from the $\mathcal{B}$ to $\mathcal{W}$ will be the corresponding negative rotations, and is given by:

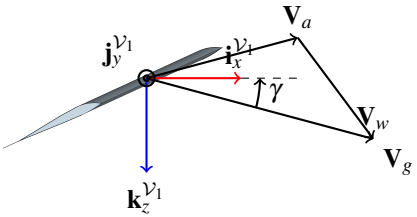
$$R_{\mathcal{B}}^{\mathcal{W}} = R_z(\beta)R_y(-\alpha) = \begin{pmatrix} c_\beta & s_\beta & 0 \\ -s_\beta & c_\beta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} c_\alpha & 0 & s_\alpha \\ 0 & 1 & 0 \\ -s_\alpha & 0 & c_\alpha \end{pmatrix} \tag{2.5}$$

Path Angle and Course Angle

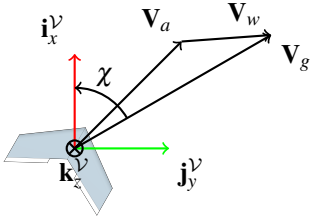
The velocity of the MAV in relation to the ground is called the ground speed vector ($\mathbf{V}_g$). The flight path angle (γ) is defined as the angle from $\mathbf{V}_g$ to its projection in the horizontal plane ($\mathbf{i}_x^{\mathcal{V}}\mathbf{j}_y^{\mathcal{V}}$-plane), see Figure 2.4a. Here we also define the course angle (χ) as the angle from the projection of the ground speed vector ($\mathbf{V}_g$) on the horizontal plane ($\mathbf{i}_x^{\mathcal{V}}\mathbf{j}_y^{\mathcal{V}}$-plane) to the north ($\mathbf{i}_x^{\mathcal{V}}$), see Figure 2.4b. The triangle created by the $\mathbf{V}_g$, $\mathbf{V}_a$ and $\mathbf{V}_w$ is usually referred to as the *wind triangle*.

2.2 MAV Modeling

In order to develop control strategies, perform simulations and getting more familiar with MAV kinematics and dynamics a MAV model was adopted from [Beard and McLain, 2012]. The states used in the full model of the MAV can be seen in Table 2.1.



(a) Path angle (γ) from a side view of the flying wing. Notice that the displayed angle represents a negative path angle.



(b) Course angle (χ) from a top-view of the flying-wing.

Figure 2.4: Illustration of course angle (χ) and path angle (γ).

MAV States		
Description	Symbol	Vector notation
MAV position in $\mathcal{I}$	$(x_m, y_m, z_m)^T$	$\mathbf{p}_m$
MAV linear velocities in $\mathcal{B}$	$(u, v, w)^T$	$\mathbf{u}$
MAV roll	ϕ	-
MAV pitch	θ	-
MAV yaw	ψ	-
MAV angular velocities in $\mathcal{B}$	$(p, q, r)^T$	ω

Table 2.1: Summary of the states used for the modeling of the MAV.

Rigid Body Kinematics

The kinematic relationships presented in this section are not specific to the MAV, but can be used for all rigid bodies using the same definitions of reference frames.The linear kinematics establish the relationship between the body frame and the inertial frame velocities of the MAV. The angular kinematics establish the relationship between the body frame angular velocities and the time derivatives of yaw, pitch and roll respectively.

Linear Kinematics The time derivative of the MAV position in $\mathcal{I}$, $(\dot{x_m}, \dot{y_m}, \dot{z_m})^T$, does not equal the body frame velocities, $(u, v, w)^T$, since they are expressed in different reference frames. The relationship can be established by rotating a vector from $\mathcal{B}$ to $\mathcal{V}$ (notice that the time derivative of a position in $\mathcal{I}$ and $\mathcal{V}$ are equal). An inverse rotation for a given rotation matrix can be performed by transposing the matrix. Thus $(R_{\mathcal{V}}^{\mathcal{B}})^T$ can be used for the purpose of this. The final relationship is given by

$$\begin{pmatrix} \dot{x_m} \\ \dot{y_m} \\ \dot{z_m} \end{pmatrix} = R_{\mathcal{B}}^{\mathcal{V}} \begin{pmatrix} u \\ v \\ w \end{pmatrix} = (R_{\mathcal{V}}^{\mathcal{B}})^T \begin{pmatrix} u \\ v \\ w \end{pmatrix} \tag{2.6}$$

Angular Kinematics The time derivative of the MAV orientation, $(\dot{\phi}, \dot{\theta}, \dot{\psi})^T$, is not equal to the body frame angular rates, $(p,q,r)^T$, as they are expressed in different reference frames. Roll(ϕ), pitch(θ) and yaw(ψ) are defined by intrinsic rotations and are therefore also internally defined in different reference frames. Using the successive rotations from the intrinsic ZYX Tait-Bryan convention (see Equation (2.3)) a relation between the two can be established as

$$\begin{pmatrix} q \\ p \\ r \end{pmatrix} = \begin{pmatrix} \dot{\phi} \\ 0 \\ 0 \end{pmatrix} + R_x(\phi) \begin{pmatrix} 0 \\ \dot{\theta} \\ 0 \end{pmatrix} + R_x(\phi) R_y(\theta) \begin{pmatrix} 0 \\ 0 \\ \dot{\psi} \end{pmatrix} \tag{2.7}$$

Multiplying matrices, extracting and concatenating columns, and inverting yields ([Beard and McLain, 2012])

$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} 1 & s(\phi)t(\theta) & c(\phi)t(\theta) \\ 0 & c(\phi) & -s(\phi) \\ 0 & s(\phi)/c(\theta) & c(\phi)/c(\theta) \end{pmatrix} \begin{pmatrix} q \\ p \\ r \end{pmatrix} \tag{2.8}$$

Rigid Body Dynamics

The dynamic relationships presented in this section are not specific to the MAV, but can be used for all rigid bodies using the same assumptions and definitions of reference frames. The dynamics establish the relationship between the time derivative of body frame linear and angular velocities, and forces and moments acting upon the body. More detailed derivations of the dynamic equations can be found in [Beard and McLain, 2012] and [Murray et al., 1994]. With a state vector

$$\mathbf{x} = \begin{bmatrix} \mathbf{u}^T & \mathbf{v}^T \end{bmatrix}^T = \begin{bmatrix} u & v & w & p & q & r \end{bmatrix}^T \tag{2.9}$$

and forces and torques as

$$\mathbf{Q} = \begin{bmatrix} \mathbf{f}^T & \tau^T \end{bmatrix}^T = \begin{bmatrix} f_x & f_y & f_z & l & \mathtt{m} & n \end{bmatrix}^T \tag{2.10}$$

where f_x, f_y and f_z are forces and l, $\mathtt{m}$ and n are moments around the $\mathbf{i}_x^{\mathcal{B}}$, $\mathbf{j}_y^{\mathcal{B}}$ and $\mathbf{k}_z^{\mathcal{B}}$ axis respectively, the dynamic equations for a rigid body can be described by the standard Newton-Euler equations below, similarly as in [Murray et al., 1994].

$$M\dot{\mathbf{x}} + C(\mathbf{x})\mathbf{x} = \mathbf{Q} \tag{2.11}$$

Here

$$M = \begin{pmatrix} mI_{3x3} & 0_{3x3} \\ 0_{3x3} & J \end{pmatrix} \tag{2.12}$$

where J is the inertia matrix and m is the mass of the MAV. Since the MAV body is symmetric in the $\mathbf{i}_x^{\mathcal{B}}\mathbf{k}_z^{\mathcal{B}}$-plane J can be expressed as

$$J = \begin{pmatrix} J_x & 0 & -J_{xz} \\ 0 & J_y & 0 \\ -J_{xz} & 0 & J_z \end{pmatrix} \tag{2.13}$$

and $C(\mathbf{x})$ is called the Coriolis matrix

$$C(\mathbf{x}) = \begin{pmatrix} m[\mathbf{u}]_\times & 0_{3x3} \\ 0_{3x3} & [\mathbf{u}]_\times J \end{pmatrix} \tag{2.14}$$

where

$$[\mathbf{u}]_\times = \begin{pmatrix} 0 & r & -q \\ -r & 0 & p \\ q & -p & 0 \end{pmatrix} \tag{2.15}$$

is the skew-symmetric vector product matrix of $\mathbf{u} = (p,q,r)^T$.

Forces and Moments

In this section will expand on the different components of the forces and moments ($\mathbf{Q}$) acting upon the MAV. The total force acting upon the MAV is the sum of gravitational forces, aerodynamic forces and propulsion forces. The total moment acting upon the MAV is the sum of aerodynamic moments and propulsion moments. For simplicity the propulsion moments will be neglected in this report. A more detailed explanation of the modeled forces and moments can be found in [Beard and McLain, 2012].

Gravitational Forces The force from gravity acting upon the MAV is

$$\mathbf{f}_g^{\mathcal{V}} = \begin{bmatrix} 0 & 0 & mg \end{bmatrix}^T \tag{2.16}$$

where m is the mass of the MAV and g is the gravitational constant. Rotating $\mathbf{f}_g^{\mathcal{V}}$ to $\mathcal{B}$ yields

$$\mathbf{f}_g^{\mathcal{B}} = R_{\mathcal{V}}^{\mathcal{B}} \mathbf{f}_g^{\mathcal{V}} \tag{2.17}$$

Control Surfaces In this report mainly two control surfaces are of interest; the elevator, mainly controlling pitch angle (θ), and the ailerons, mainly controlling roll angle (ϕ). A positive elevator deflection angle (δ_e) is defined as a positive rotation of the elevator joint around the $\mathbf{j}_y^{\mathcal{B}}$-axis (trailing edge pointing down). An aileron deflection angle (δ_a) defined as

$$\delta_a = \frac{1}{2}(\delta_{a,left} - \delta_{a,right}) \tag{2.18}$$

where the deflection angle $\delta_{a,left}$ is positive with the trailing edge pointing down and the deflection angle $\delta_{a,right}$ is positive with the trailing edge pointing up.

A flying wing has neither rudder nor elevator control surfaces. Lacking a rudder, direct control of yaw (ψ) is not possible for a flying wing. Instead of an elevator control surface, the ailerons of a flying wing, usually called elevons, are simultaneously used as ailerons and elevator. The deflection angles of the right elevon (δ_{er}) and the left elevon (δ_{el}) are both defined positive when their trailing edge is pointing down (see Figure 2.5). With these definitions we can convert between elevator and aileron commands to elevon commands according to

$$\begin{pmatrix} \delta_e \\ \delta_a \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix} \begin{pmatrix} \delta_{er} \\ \delta_{el} \end{pmatrix} \tag{2.19}$$

With this conversion established the aerodynamic forces and moments can be expressed in terms of δ_e and δ_a [Beard and McLain, 2012] (here there is an error in the original publication, corrected in the errata [*Errata; Small Unmanned Aircraft: Theory and Practice* 2020]).

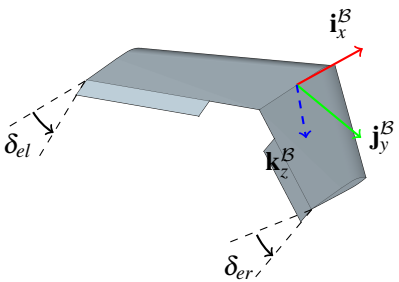


Figure 2.5: Illustration of the left and right elevon deflection angles (δ_{el} and δ_{er}).

Aerodynamic Forces and Moments The longitudinal aerodynamic forces and moments are expressed in the stability frame ($\mathcal{S}$). The forces and moments modeled are lift force (F_{lift}) along the negative $\mathbf{k}_z^{\mathcal{S}}$-axis, drag force ($F_{drag}$) along the negative $\mathbf{i}_x^{\mathcal{S}}$-axis and moment $\mathtt{m}$ (see Figure 2.6). In order to express forces in $\mathcal{B}$, F_{lift} and F_{drag} can be converted from $\mathcal{S}$ by negation and then rotation by α [Beard and McLain, 2012].

$$F_{lift} = \frac{1}{2}\rho V_a^2 S C_L(\alpha, q, \delta_e) \tag{2.20}$$

$$F_{drag} = \frac{1}{2}\rho V_a^2 S C_D(\alpha, q, \delta_e) \tag{2.21}$$

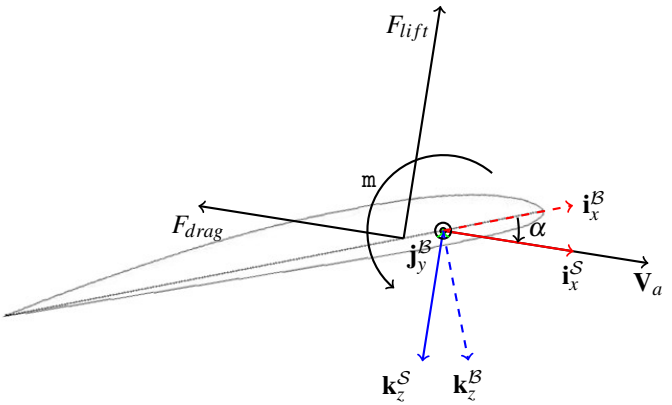


Figure 2.6: Lift and drag forces in relation to frames $\mathcal{S}$ and $\mathcal{B}$. Notice that the forces are acting upon the aerodynamic center rather than center of gravity. This also introduces the moment $\mathtt{m}$ around $\mathbf{j}_y^{\mathcal{B}}$.

$$\mathtt{m} = \frac{1}{2}\rho V_a S c C_{\mathtt{m}}(\alpha, q, \delta_e) \tag{2.22}$$

where ρ is the air-density, S is the planform area of the MAV, c is the mean chord of the MAV wing and, C_L, C_D and $C_{\mathtt{m}}$ are aerodynamic coefficients depending on α, q and δ_e. The nonlinear aerodynamic coefficients, C_L, C_D and $C_{\mathtt{m}}$, are approximated with a first order Taylor series expansions.

Notice that F_{lift} is highly nonlinear in angle of attack (α), specially when the magnitude of α is large. In order to model stall behaviour this non-linearity has to be taken into account. Thus, the linear α-dependency in the Taylor expansion is substituted with a nonlinear function. F_{drag} is also nonlinear in angle-of-attack and this α-component is in a similar way substituted with a nonlinear function. Finally

$$C_L(\alpha, q, \delta_e) = C_L(\alpha) + C_{L_q}\frac{c}{2V_a}q + C_{L_{\delta_e}}\delta_e \tag{2.23}$$

$$C_D(\alpha, q, \delta_e) = C_D(\alpha) + C_{D_q}\frac{c}{2V_a}q + C_{D_{\delta_e}}\delta_e \tag{2.24}$$

$$C_{\mathtt{m}}(\alpha, q, \delta_e) = C_{\mathtt{m}_0} + C_{\mathtt{m}_\alpha}\alpha + C_{\mathtt{m}_q}\frac{c}{2V_a}q + C_{\mathtt{m}_{\delta_e}}\delta_e \tag{2.25}$$

where the parameters are usually found from wind tunnel experiments or computer fluid dynamics (CFD) simulation, and

$$C_L(\alpha) = (1-\sigma(\alpha))[C_{L_0} + C_{L_\alpha}\alpha] + \sigma(\alpha)[2sign(\alpha)sin^2(\alpha)cos(\alpha)] \tag{2.26}$$

where $\sigma(\alpha)$ is a sigmoid blending function and

$$C_{L_\alpha} = \frac{\pi b^2/S}{1+\sqrt{1+(AR/2)^2}} \tag{2.27}$$

where b is the wingspan and

$$C_D(\alpha) = C_{D_p} + \frac{(C_{L_0}+C_{L_\alpha})^2}{\pi e b^2/S} \tag{2.28}$$

where C_{D_p} is the parasitic drag and e is the Oswald efficiency factor [Beard and McLain, 2012].

The lateral aerodynamic forces and moments are expressed in the body frame. The forces and moments modeled are f_y, l and n.

$$f_y = \frac{1}{2}\rho V_a^2 S C_Y(\beta, p, r, \delta_a) \tag{2.29}$$

$$l = \frac{1}{2}\rho V_a^2 S b C_l(\beta, p, r, \delta_a) \tag{2.30}$$

$$n = \frac{1}{2}\rho V_a^2 S b C_n(\beta, p, r, \delta_a) \tag{2.31}$$

where b is the wingspan of the MAV and, C_Y, C_l and C_n are aerodynamic coefficients depending on β, p, r and δ_a.The parameters C_L, C_D and C_m are approximated with a first order Taylor series expansion, similar as for the longitudinal components, but without any nonlinear behaviour except for the V_a dependency. Finally

$$C_Y(\beta, p, r, \delta_a) = C_{Y_0} + C_{Y_\beta}\beta + C_{Y_p}\frac{c}{2V_a}p + C_{Y_r}\frac{c}{2V_a}r + C_{Y_{\delta_a}}\delta_a \tag{2.32}$$

$$C_l(\beta, p, r, \delta_a) = C_{l_0} + C_{l_\beta}\beta + C_{l_p}\frac{c}{2V_a}p + C_{l_r}\frac{c}{2V_a}r + C_{l_{\delta_a}}\delta_a \tag{2.33}$$

$$C_n(\beta, p, r, \delta_a) = C_{n_0} + C_{n_\beta}\beta + C_{n_p}\frac{c}{2V_a}p + C_{n_r}\frac{c}{2V_a}r + C_{n_{\delta_a}}\delta_a \tag{2.34}$$

where the parameters are usually found from wind tunnel experiments or computer fluid dynamics (CFD) simulation.

Propulsion Using Bernoulli's equation, the thrust force ($\mathbf{f}_p^{\mathcal{B}}$) can be expressed by the pressure difference upstream and downstream of the propeller [Beard and McLain, 2012]. The final propulsion force is modeled as

$$\mathbf{f}_p^{\mathcal{B}} = \frac{1}{2}\rho S_{prop} C_{prop} \begin{pmatrix} (k_{motor}\delta_t)^2 - V_a^2 \\ 0 \\ 0 \end{pmatrix} \tag{2.35}$$

where S_{prop} is the area swept by the propeller, C_{prop} is a propeller coefficient and k_{motor} is a constant defining the proportional relationship between thrust command δ_t to the speed of the airflow exiting the motor.

Parameters

Currently there is no available set of aerodynamic parameters for the designed flying-wing MAV intended for usage in the real-world mission. Instead, a slightly modified version of the parameter-set for a flying-wing MAV called Zagi, identified in [Platanitis and Shkarayev, 2005] was used. The Zagi MAV is slightly larger than the intended MAV, and can be seen in Figure 2.7 [*Zagi website* 2020]. The full set of parameters used for the simulation can be seen in Appendix A.

Figure 2.7: The Zagi flying-wing. [*Zagi website* 2020]

Simplified MAV Model

For control design a simplified model of the process might be beneficial to use. For this purpose, a slightly modified version of the model presented in [Persson, 2019] is used. In the original model the direction of flight is set by the flight path angle (γ) and the course angle (χ) (see Section 2.1). For this model it is also assumed that the acceleration (a), path angle (γ) and course angle (χ) are set by some low-level controller with first order, unit gain transfer functions. The simplified model of the MAV can be seen below

$$\begin{bmatrix} \dot{x}_s \\ \dot{y}_s \\ \dot{z}_s \\ \dot{v}_s \\ \dot{a}_s \\ \dot{\gamma}_s \\ \dot{\chi}_s \end{bmatrix} = \begin{bmatrix} v_s cos(\gamma_s) cos(\chi_s) \\ v_s cos(\gamma_s) sin(\chi_s) \\ -v_s sin(\gamma_s) \\ a_s \\ \frac{1}{\tau_a}(a_{s_{cmd}} - a_s) \\ \frac{1}{\tau_\gamma}(\gamma_{s_{cmd}} - \gamma_s) \\ \frac{1}{\tau_\chi}(\chi_{s_{cmd}} - \chi_s) \end{bmatrix} \tag{2.36}$$

where v_s is the speed, $a_{s_{cmd}}$, $\gamma_{s_{cmd}}$ $\chi_{s_{cmd}}$ are inputs and τ_a, τ_γ and τ_χ are time constants. Since a flying wing can not directly control yaw (lacking a rudder), yaw and course angle are controlled by rolling the MAV, which is having the secondary effect of changing the yaw. Somewhat simplified, the ailerons control the roll angular rate, integrating up to roll angle. In turn, roll is affecting the yaw-rate, integrating up to yaw and course angle. Thus, the behaviour of the course angle to some degree resembles a second order transfer function. Modeling this behaviour, the χ_s transfer function is extended to a second order function with unit gain, on the form

$$\chi_s = \frac{\tau_{\chi_0}}{s^2 + \tau_{\chi_1} s + \tau_{\chi_0}} \chi_s^{cmd} \tag{2.37}$$

Converting Equation (2.37) to state space form we get the final simplified model below

$$\begin{bmatrix} \dot{x}_s \\ \dot{y}_s \\ \dot{z}_s \\ \dot{v}_s \\ \dot{a}_s \\ \dot{\gamma}_s \\ \dot{\chi}_s \\ \ddot{\chi}_s \end{bmatrix} = \begin{bmatrix} v_s cos(\gamma_s) cos(\chi_s) \\ v_s cos(\gamma_s) sin(\chi_s) \\ -v_s sin(\gamma_s) \\ a_s \\ \frac{1}{\tau_a}(a_{s_{cmd}} - a_s) \\ \frac{1}{\tau_\gamma}(\gamma_{s_{cmd}} - \gamma_s) \\ \dot{\chi}_s \\ -\tau_{\chi_1}\dot{\chi} + \tau_{\chi_0}(\chi_s^{cmd} - \chi_s) \end{bmatrix} \tag{2.38}$$

where τ_{χ_0} and τ_{χ_1} should be chosen to make the model dynamics as similar as possible to the real MAV dynamics.

2.3 Wind Modeling

As mentioned in Section 2.1, the ground velocity, $\mathbf{V}_g$, of the MAV is the sum of the air speed vector, $\mathbf{V}_a$, and the wind velocity $\mathbf{V}_w$. This relation is usually called the wind triangle.

$$\mathbf{V}_g = \mathbf{V}_a + \mathbf{V}_w \tag{2.39}$$

The modeled wind velocity consists of two parts, one ambient stationary component and one stochastic gust component

$$\mathbf{V}_w = \mathbf{V}_{w_a} + \mathbf{V}_{w_g} \tag{2.40}$$

where $\mathbf{V}_w$ is the total wind velocity, $\mathbf{V}_{w_a}$ is the ambient component and $\mathbf{V}_{w_g}$ is the gust component. The ambient component is defined in the inertial frame and the gust component is defined in the body frame

$$\mathbf{V}_{w_a}^{\mathcal{I}} = (w_{x_a}, w_{y_a}, w_{z_a})^T \tag{2.41}$$

$$\mathbf{V}_{w_g}^{\mathcal{B}} = (u_{w_g}, v_{w_g}, w_{w_g})^T \tag{2.42}$$

Dryden Wind Model

The gust component of the wind is modeled with a Dryden wind model. The Dryden model is defined by one transfer function for each gust component. The gust is the output from the transfer function when filtering white noise. The transfer function for each component is [Beard and McLain, 2012]

$$H_u(s) = \sigma_u \sqrt{\frac{2V_a}{L_u}} \frac{1}{s + \frac{V_a}{L_u}} \tag{2.43}$$

$$H_v(s) = \sigma_v \sqrt{\frac{3V_a}{L_v}} \frac{(s + \frac{V_a}{\sqrt{3}L_v})}{(s + \frac{V_a}{L_v})^2} \tag{2.44}$$

$$H_w(s) = \sigma_w \sqrt{\frac{3V_a}{L_w}} \frac{(s + \frac{V_a}{\sqrt{3}L_w})}{(s + \frac{V_a}{L_w})^2} \tag{2.45}$$

where σ_u, σ_v and σ_w are the intensity for each component of the white noise driving the process and L_u, L_v and L_w are spatial wavelengths.

2.4 Boat Modeling

In order to perform full system simulations, a five state boat model is used. The model is based on a planar 2D model with simple boat kinematics. The model does not exert any pitch or roll behaviour and the the elevation z_b is driven by the behaviour of the waves, described in Section 2.5.

$$\begin{bmatrix} \dot{x_b} \\ \dot{y_b} \\ z_b \\ \dot{\psi_b} \\ \dot{v_b} \end{bmatrix} = \begin{bmatrix} v_b cos(\psi_b) \\ v_b sin(\psi_b) \\ P_w(x_b, y_b, t) \\ v_b \frac{1}{d_r} tan(k_r \delta_r) \\ \frac{1}{m_b}(-C_d v_b^2 + k_t \delta_u) \end{bmatrix} \tag{2.46}$$

where δ_r and δ_u are input signals, m_b is the mass of the boat, x_b, y_b and z_b is the boat position in $\mathcal{I}$, v_b is the speed and ψ_b is the heading. C_d is a coefficient for modeling the drag of the water. The steering behavior is defined by d_r, the distance between the center of gravity and the center of rotation of the rudder, the steering input rudder angle δ_r and a steering coefficient k_r. The model behavior is decreasingly realistic with increased rudder angle.

The physical boat parameters have been taken from the "Viktoriaklassen" type boat, which is a commonly used boat in the mid size range in the SSRS fleet. A few characteristics are [*SSRS rescue fleet* 2020].

- **Mass**: 13 tonnes
- **Max velocity**: 34 knots = 17.5 m/s

The coefficient of drag C_d is set such that a full throttle ($\delta_u = 1$) results in the steady state maximum velocity of 17.5 m/s.

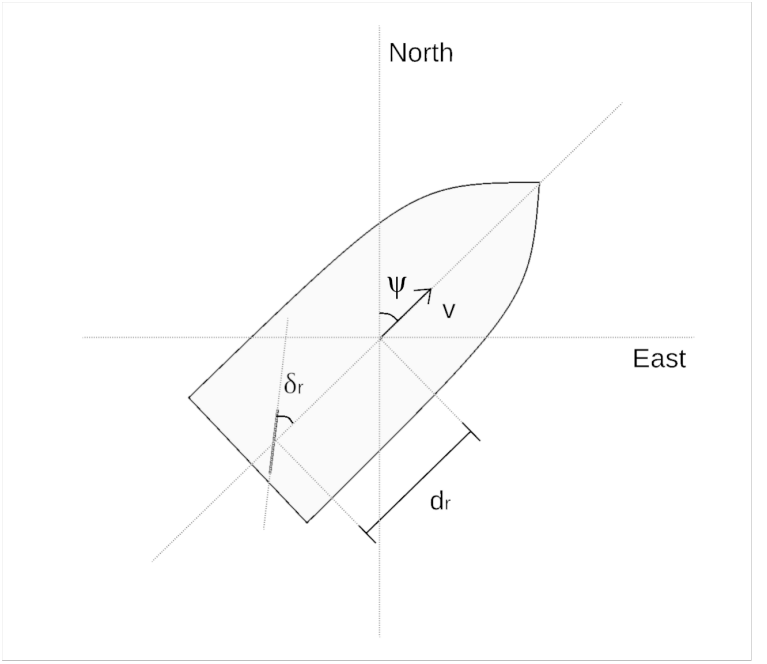


Figure 2.8: Simplified kinematic boat model.

Simplified Boat Model

An even simpler boat model will be used as a part of the control algorithm. The boat model presented above could have been used here, but the simplified model is meant to showcase the modeling error between the boat plant and the internal model used by the control algorithm. For this purpose we use a unicycle, sometimes called *differential drive*, as a minimal boat model. The system dynamics are

$$\begin{bmatrix} \dot{x_b} \\ \dot{y_b} \\ \dot{\psi_b} \end{bmatrix} = \begin{bmatrix} \delta_v cos(\psi_b) \\ \delta_v sin(\psi_b) \\ \delta_\psi \end{bmatrix} \tag{2.47}$$

2.5 Wave Modeling

The waves are modeled as a displacements along $\mathbf{k}_z^{\mathcal{I}}$ from the nominal sea level. Just as for the wind, the model consists of two parts, one deterministic component and one stochastic component.

$$P_w = P_{w_d} + P_{w_s} \tag{2.48}$$

where P_w is the total wave displacement, P_{w_d} is the deterministic component and P_{w_s} is the stochastic component. The deterministic component is modeled as

$$P_{w_d}(x_b, y_b, t) = Acos\left(\frac{2\pi cos(\psi_{wave})}{\lambda} x_b + \frac{2\pi sin(\psi_{wave})}{\lambda} y_b - \frac{2\pi t}{T} + \phi_{wave}\right) \tag{2.49}$$

where A is the wave amplitude, λ is the wavelength, T is the wave period, ψ_{wave} is the direction of the wave, ϕ_{wave} is the phase of the wave and t is the current instance in time. For now the stochastic component is modeled as white noise.

3

Positioning

This chapter explores different relative positioning methods between MAV and boat. Different configuration options of the sensor solutions are presented. General requirements and robustness aspects are presented and discussed. Several potential positioning solutions are explored and evaluated. Tests are performed on physical devices. Finally, two of the the positioning systems are modeled to be used in a full system simulation replicating the intended target scenario to evaluate their performance. Discussions directly related to tests on the positioning systems or subsystems of them are made in this chapter. More general discussions related to the performance of the positioning systems in the simulation environment together with the control systems are presented in Chapter 8.

3.1 Main Robustness Aspects

In order to make the comparison between different positioning solutions more manageable, the robustness aspects of a positioning system are divided into a number of categories.

State Information In order to be able to control the MAV towards the boat, relevant information has to be made available to the controller. The positioning system can output different quantities, an absolute or a relative position measurement depending on the sensor system. The positioning system output can also express the measurement in different coordinate frames, which may or may not be easily transformable to an appropriate state for the control system, which is why each positioning system output has to be evaluated with the final control system in mind.

Performance There are several performance related aspects of a positioning system and in this thesis, the most relevant ones that have been considered are accuracy, precision, refresh rate and delay. Accuracy and precision determine how well the measurement corresponds with the real position state. Accuracy describes the statistical bias of the measurements and the precision describes the statistical variability of the measurements. Accuracy and precision can for example be affected by

factors such as sensor hardware, sensor mounting, signal noise and detection range. The delay and refresh rate of the positioning system are additional performance factors and can cause instability problems among others.

Environment The SSRS rescue missions occur in many weather conditions. Specifically, the system has to work in rainy and windy conditions. In addition, it has to function in different light conditions, including everything from night time darkness to daytime sunlight as well as sun glare in the water. Every year on most parts of the Swedish west coast, there are between 100 and 150 days with rain, 80 to 100 days with at least 1 mm of rain, and 20 to 30 days with at least 10 mm of rain [*SMHI - Nederbörd* 2020]. However, accidents tend to occur more frequently on days without rain, as people in general are more likely to go out with their boats those days. Based on statistical data from SSRS, 5% of incidents occur when winds are stronger than 15 m/s, which is an indication that strong winds can be expected.

Interaction Some positioning systems require data-communication while others can operate differently, for example by radio wave interference patterns. The interaction aspect is related to the connection between the MAV and the positioning system, what is needed to establish a connection and how reliable the connection is.

Simplicity It is beneficial if the positioning system is kept as simple as possible since the development of SSRS' work with autonomous systems partially is carried out through university projects or volunteer work. The person who implements or builds something, might not be the one operating or providing maintenance on the system at a later point in time.

Weight Addition

For some relative positioning systems there is a need to implement a sensor device in the MAV, which potentially could add weight, volume and claim power consumption. Weight addition to the MAV will increase the energy usage and shorten the maximum possible time spent in the air. To evaluate the effect of added weight on flight time, the lost flying time can be expressed as a function of added weight. The design criteria of the custom designed airframe referenced in Appendix C was that the drone should be able to fly at 35 m/s for 10 minutes and then loiter at the scene of incident for at least 40 minutes. Assuming that the first flight phase of 10 minutes at 35 m/s remains at 10 minutes and that the decreased time only gets subtracted from the second flight phase, the total energy required to perform the flight can be approximated by

$$E_b = P_1 t_1 + P_2 t_2 \tag{3.1}$$

where E_b is the battery energy required and P and t are the power and time for the two different phases of flight. From the SSRS airframe design study, a motor and propeller efficiency of 60 % is assumed for power calculations and the electronics

on board is assumed to consume 5W continuously. The power consumption at a certain point is then given by

$$P = (F_d V_a)\frac{1}{0.6} + 5 \tag{3.2}$$

where V_a is the airspeed and F_d is the drag force. The drag force is a function of the airspeed and the angle of attack. For steady state flight, the MAV adapts the angle of attack to achieve a net zero vertical force component. Added weight has to be compensated by an increase in the angle of attack to maintain this condition, which in turn increases the drag force. To get an approximate value on how the drag forces are affected by the mass, the MAV model, referenced in Section 2.2 was simulated in steady state flight at 15 m/s and 35 m/s and the drag force was logged for a variety of mass values. The simulations showed a 1.0 mN increase in added drag force per added gram at 15 m/s and 7.6 mN at 35 m/s. This corresponds to about 9 seconds of lost loitering flight time for every added gram of weight.

3.2 Possible Sensor Configurations

Since relative position can be found using many different sensor solutions, there are multiple ways they can be physically mounted and configured. For example, a vision based system can consist of a camera mounted on the MAV, on the boat, or on both. Different configurations require different considerations. For the target scenario, the possible configurations was divided into three major categories explained below.

Active Boat, Passive MAV In this configuration, the positioning is performed by a sensor solution on the boat. The benefit of this setup is that no additional sensor on the MAV is required which limits the risk of additional weight, space or power consumption. The drawback of this configuration is that a communication link is required to feed information between the boat and the MAV.

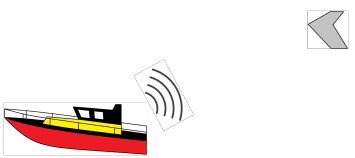


Figure 3.1: Active boat, passive MAV

Passive Boat, Active MAV In this configuration, the positioning is performed by a sensor solution on the MAV. The benefit of this configuration is that the position yielded is instantly ready to be processed for control output without the need for a communication link. The drawback of this configuration is that the sensor solution adds weight, volume and power consumption.

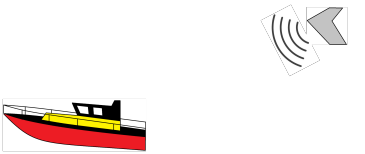


Figure 3.2: Passive boat, active MAV

Active Boat, Active MAV This configuration uses sensor solutions on both vehicles. This could potentially improve the accuracy of the relative position measurement and provide measurement redundancy. Although, the drawbacks of this configuration are both of the previous two configurations; added weight and more space required on the MAV as well as the need for a communication link.

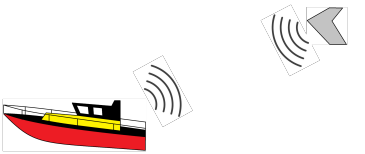


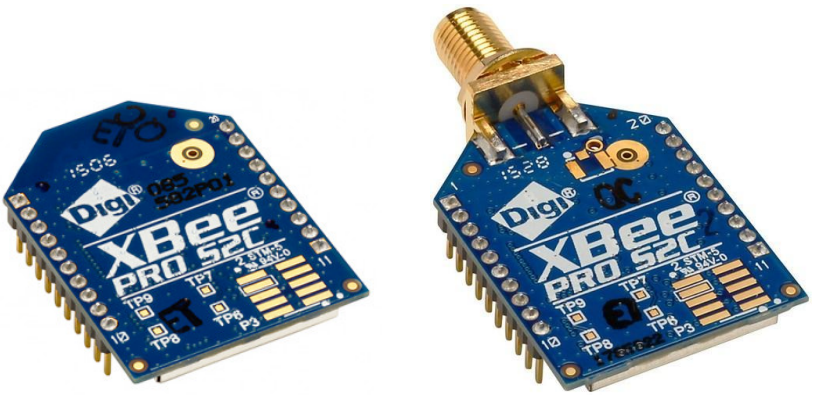
Figure 3.3: Active boat, active MAV

3.3 Communication

In the sensor configurations where the boat actively retrieves measurements, there is a need for data to be communicated to achieve closed loop control. This can principally be performed in many ways, but a radio communication link provides a commercially available and relatively easy way to transfer data. In the long run, the communication has to work between boats and devices in such a way that any boat arriving at any scene of incident should be able to communicate with any one MAV sent out to the scene. Initializing and maintaining the communication link has to be robust and without the need of manual setup.

In order to evaluate one radio communication device, an Xbee RF-module (2.4 GHz), as seen in Figure 3.4, was chosen. The reason for choosing this device was because it has multiple antenna configuration options. This allows the lightweight (4 g) version, seen in Figure 3.4a, with the antenna integrated into the Printed Circuit Board (PCB) to be used one the MAV side. On the boat side, the version with the standardized SMA antenna connector, seen in Figure 3.4b, allows any suitable antenna that uses the SMA connector type to be used. Aside from the antenna aspect, the protocol used in Xbee supports a broadcast mode which makes it possible to communicate with any device in a predefined network. This makes the protocol

appropriate for the task as it makes any boat compatible with any MAV as well as the ability for any boat to communicate with multiple drones.



(a) Xbee with PCB antenna.

(b) Xbee with SMA antenna connector.

Figure 3.4: The different Xbee antennas used for radio communication.

Testing

To test the range of the device, multiple combinations of devices were used. Firstly, two devices with PCB antennas were separated slowly while communicating, and the distance at which they stopped communicating was noted. The same procedure was performed for the combination PCB antenna and WiFi dipole antenna. The results can be seen in Table 3.1.

Antenna	PCB - PCB	PCB - Dipole
Maximum range	35 m	550 m

Table 3.1: Test results from the range tests of Xbee communication modules with different antennas.

In order to evaluate the magnitude and variation of the communication delay, 2000 messages each containing 14 bytes of data were sent from one device to another and then back again. The round trip time was saved and divided by two to get an approximation of the one way delay time. The 14 bytes represent approximately the amount of data required to send one position measurement containing 5 bytes each for a latitude and longitude and 3 bytes for altitude. One extra byte was added as margin. As seen in Figure 3.5, the delay is normally below 23 ms. Two distinct peaks can be seen and the cause of this phenomenon is not clear. A possible explanation is that the low level communication protocol has a fault detection function which is resending messages when data arrives corrupt, and thus sometimes adding time.

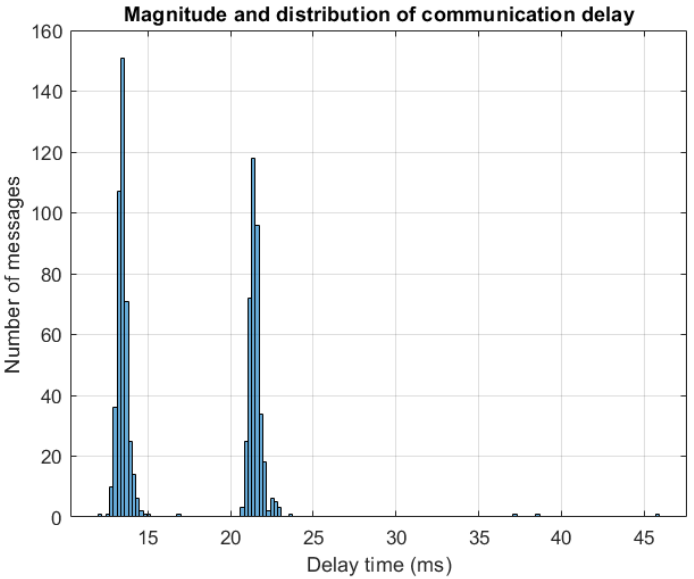


Figure 3.5: Magnitude and distribution of delay for 2000 messages sent between devices.

3.4 Selection of Positioning Systems

Three major positioning system alternatives have been evaluated. The systems are Global Navigation Satellite System (GNSS) based positioning, 3D LiDAR based positioning and a guidance system based on the Instrument Landing System (ILS). GNSS positioning was chosen because the MAV will be equipped with a module for other navigational purposes than landing, and utilizing it for landing would make it possible to achieve the goal without the use of an additional positioning module. This can in turn save weight and space. 3D LiDAR was chosen as an alternative because commercially available modules today provide a relatively accurate and fast measurement. When mounted in the "Active boat - passive MAV" configuration, it also does not add weight and extra power consumption on the MAV. 3D LiDARs are becoming popular due to use in other industries which is likely to increase the availability of high performance devices in the future. An Instrument Landing System (ILS) inspired solution using Received Signal Strength Indication (RSSI) and directional antennas was chosen to be explored because of the fact that it would not explicitly require any data communication which reduces the complexity. Depending on the measurement principle, this method also has potential to be a lightweight solution.

3.5 GNSS Positioning

Global Navigation Satellite System (GNSS) positioning is a common way to measure position for navigation purposes. GNSS refers to a collection of satellite constellations that mainly consist of the Global Positioning System (GPS) (USA), GLONASS (Russia), Galileo (Europe) and BeiDou (China) [*Other Global Navigation Satellite Systems (GNSS)* 2020].

The working principle of a GNSS module is based on distance measurements between a receiver and multiple satellites in a constellation. The distance measurement is done by measuring the time difference between a message to a certain satellite being sent and received, which can be multiplied by the speed of light to retrieve a pseudorange. Through trilateration of multiple pseudorange measurements, a position can be estimated. When measuring the time of flight of a GNSS signal, the system uses the code phase. This works by letting the GNSS device send out a series of randomized patterns which is received and returned back by a satellite. The GNSS device continuously checks the correlation between the signal patterns of the transmitted signal the received ones and when a surge in correlation is detected, the time difference can be determined. Some GNSS devices also utilize the carrier phase to increase the accuracy of the time measurement. This works by measuring the phase of the carrier signal with which the message code is sent. Since the carrier frequency is much larger than the message frequency, the resolution of the measurement is increased.

Positioning systems based on GNSS suffer from errors that affect the accuracy of the measurement. The type of errors can be divided into three categories; constant errors, correlated errors and uncorrelated errors. [Karaim et al., 2018] [Acharya, 2014a]

Constant Errors The constant error sources stem from the satellite and affect every receiver it interacts with in a similar way. These errors can be caused by delay in the satellite hardware or bias in its clock.

Correlated Errors The correlated errors are regional phenomenons that affect GNSS receivers that are positioned in proximity to each other similarly. These errors can be caused by the propagation of the radio signal, for example by errors in the ionosphere and troposphere. These errors vary with the geographic location of the receiver but can generally be well correlated in one region.

Uncorrelated Errors Uncorrelated errors are independent of geographic location. This category contains signal noise errors caused by nearby devices, and multipath errors, which is caused by reflection of the signal on objects and terrain in the surrounding area of the receiver.

For GNSS positioning, it is possible to send correction data between devices to achieve better precision. One way to achieve this is called Real-Time Kinematic (RTK). This method is based on having a stationary base station at a known position that reports the measurement deviation in real-time to one or multiple rovers in a receiver constellation. There also exists ways of similarly sending correction data between multiple rovers but without the base station, for the purpose of achieving better relative position when high precision absolute position is irrelevant [Acharya, 2014b]. This type of RTK integrated GNSS positioning is not considered in this report, as the RTK method requires a high rate data communication between the devices, which can be argued to add unnecessary complexity to a system that is intended to be kept to a minimum.

When using two GNSS receivers for relative position, the precision of the measurement depends on how large the constant and correlated errors are in relation to the uncorrelated error. The position measurement retrieved from a GNSS receiver can be expressed as

$$\mathbf{p}_r^g = \mathbf{p}_r + \mathbf{w}_r \tag{3.3}$$

where $\mathbf{p}_r^g$ is the measured GNSS receiver position, $\mathbf{p}_r$ is the actual receiver position and $\mathbf{w}_r$ is the measurement error, all expressed in the inertial frame according to

$$\mathbf{p}_r^g = \begin{bmatrix} x_r^g & y_r^g & z_r^g \end{bmatrix}^T, \qquad \mathbf{p}_r = \begin{bmatrix} x_r & y_r & z_r \end{bmatrix}^T, \qquad \mathbf{w}_r = \begin{bmatrix} w_x & w_y & w_z \end{bmatrix}^T$$

The relative measured GNSS position of a MAV and boat can therefore be expressed as

$$\Delta\mathbf{p}^g = \mathbf{p}_b^g - \mathbf{p}_m^g = \mathbf{p}_b - \mathbf{p}_m + \mathbf{w}_b - \mathbf{w}_m \tag{3.4}$$

where subindex b is used for the boat and subindex m for the MAV. As seen in Equation (3.4), the measurement precision depends on the size of the terms $\mathbf{w}$ and how well they correlate, which depends on the relative size of the constant and correlated error compared to the uncorrelated error. In the subsection "Testing", this is explored further.

Devices

In the market segment of small, lightweight GNSS modules for aerial vehicles and drones, there are a variety of devices available. Many devices are based on u-Blox chips, two of which are studied in this thesis; u-Blox NEO-M8N and u-Blox ZED-F9P. These GNSS modules can be seen in Figure 3.6. The main reason for comparing the two is to evaluate the difference in performance and whether the best

achievable performance is enough for the task. The NEO-M8N is currently used by SSRS for testing. This device uses only one frequency band, referred to as L1, while the ZED-F9P uses two, L1 and L2. The L2 band uses a higher frequency than L1 and devices that implement this band can be expected to achieve higher performance. The specifications of the two devices can be seen in Table 3.2.

Figure 3.6: The two GNSS modules. NEO-M8N on the left and ZED-F9P with its Helix antenna on the right.

Device	NEO-M8N	ZED-F9P
Frequency bands	L1	L1 & L2
Specified precision[1]	2.5 m	1.0 m
Max. refresh rate	5 Hz	20 Hz
Weight (incl. antenna)	16 g	24 g
Dimensions	47 x 42 x 10 mm	103 x 32 x 28 mm

Table 3.2: Comparison of the two GNSS devices. [*u-Blox NEO-M8 datasheet* 2020] [*u-Blox ZED-F9P datasheet* 2020]

Implementation

In order to achieve the necessary measurements and communication, the setup have be configured according to "active MAV - active boat", thus requiring a data communication link between them. A schematic of how this could be set up can be seen in Figure 3.7.

Testing

To evaluate how well two concurrent position measurements correlate, a series of relative position tests were conducted. One test was performed with two NEO-M8N

[1] Precision according to Circular Error Probable (CEP), defined as the radius of a circle within which 50% of all stationary measurements occur during a 24h test [*u-Blox ZED-F9P datasheet* 2020].

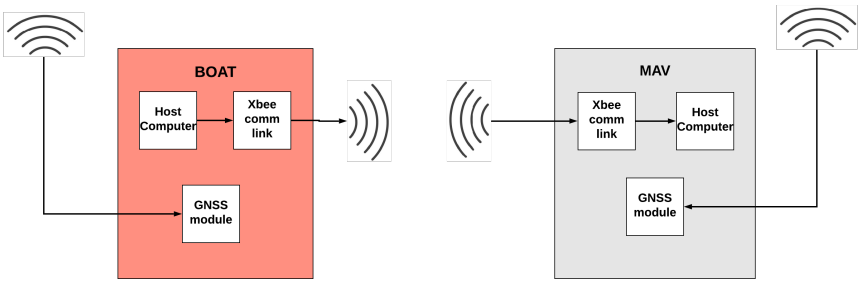


Figure 3.7: Chart of the implementation of the relative GNSS positioning system.

devices and one was performed with two ZED-F9P devices. In order to replicate the environment where the positioning system is intended to be used, the test was performed at a long jetty in Bjärred, Skåne. Using this location, the site for the test was surrounded by water and far away from large objects which could possibly cause multipath errors that would not be prevalent in the target scenario.

Setup The test was performed for both a pair of NEO-M8N receivers and a pair of ZED-F9P receivers. For each test, the receiver pair was placed stationary approximately 20 m apart from each other on the jetty, as seen in Figure 3.8a. Each receiver was connected to a Raspberry Pi host computer and the Xbee communication link. The position was recorded with a the maximum frequency of the devices (see Table 3.2). One of the receivers streamed its position in real time to the host computer of the other receiver, where both positions were recorded during a period of 15 minutes. As mentioned in Section 3.3, the Xbee communication adds approximately 20 ms delay to the "boat" position measurement, which was not compensated for.

(a) Test site at Långa bryggan, Bjärred.

(b) Setup of one of the u-Blox NEO-M8N GNSS receivers during the test.

Figure 3.8: The test location and setup for the GNSS tests.

Results The test results consists of the absolute position deviations compared to the relative position deviation for the two device pairs (see Figure 3.9-3.11). In the plots, the deviation from the average measured position is shown for each device and device pair. The numerical results are summarized in Table 3.3 and contains values of average and maximum measured position deviation for absolute and relative measurements for each device as well as the correlation for each device pair. The measurements for the NEO-M8N device can be seen in Figure 3.9.

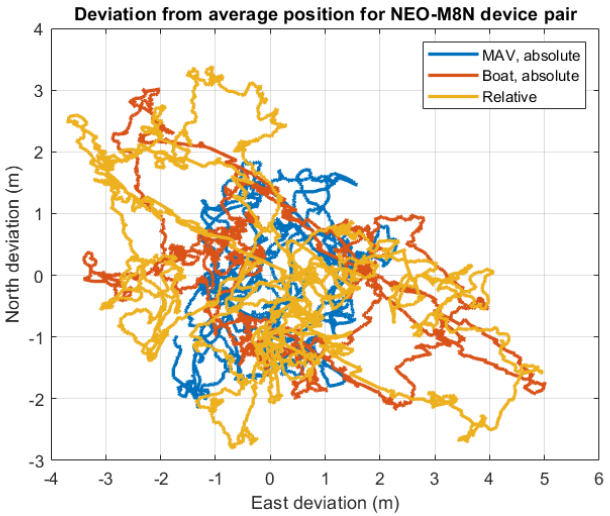


Figure 3.9: Deviation over time for the measured absolute position shown in blue and red, and the relative position, shown in yellow.

The measurements for the ZED-F9P device can be seen in Figure 3.10.
A comparison of the relative position deviation between the two different models can be seen in Figure 3.11.

	NEO-M8N	ZED-F9P
Average deviation from absolute position (m)	1.52	0.31
Maximum deviation from absolute position (m)	5.31	0.77
Average deviation from relative position (m)	2.12	0.18
Maximum deviation from relative position (m)	5.21	0.43
Average correlation between a receiver pair	0.02	0.72

Table 3.3: Summary and comparison of values on the measurement deviation for the two device pairs. The correlation value is the average of the longitude and latitude correlations for each device pair.

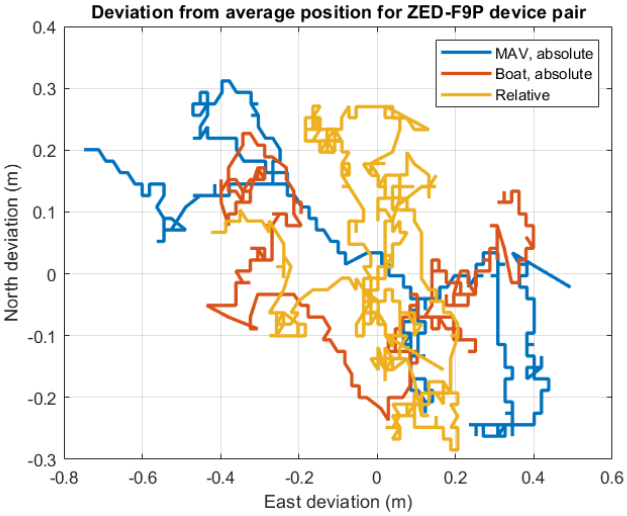


Figure 3.10: Deviation over time for the measured absolute position shown in blue and red, and the relative position, shown in yellow.

The error of the relative position measurement is 39% larger than for the absolute position measurement for the NEO-M8N. For the ZED-F9P, the relative position error is 42% smaller than for the absolute position measurement.

Discussion The results of the test show that not only is the absolute position deviation of the ZED-FP9 significantly smaller than that of the NEO-M8N, but the correlation of the measurements of the receiver pair is significantly larger, making the relative position measurement even better. For the NEO-M8N device it can be seen that the average error is increased by 39% for relative position compared to the absolute position. The average error is decreased by 42% for the ZED-F9P, which indicates that the ratio of constant and correlated measurement error to uncorrelated error is larger for the ZED-F9P receiver. Aside from the higher general performance of the ZED-F9P, this is likely to partially be a result of a better antenna. The error of the ZED-F9P is small and shows promising results. It should be noted that the test was performed during 15 minutes and that a longer test could yield a larger error.

Modeling

To model the behavior of the GNSS modules in simulations, the relative position error from each device pair retrieved during testing was used as a relative position error in the simulation model. Every simulation attempt had a randomized starting point in the error time series to better simulate the unpredictability of the measurement. The delay caused by the communication link was also added.

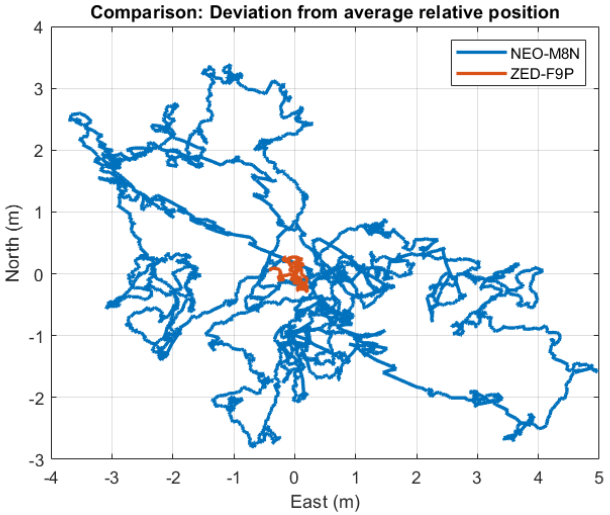


Figure 3.11: Deviation from average relative position over time for the two different GNSS devices, NEO-M8N in blue, and ZED-F9P in orange.

3.6 3D LiDAR

LiDAR is a distance measuring method that uses the time-of-flight (ToF) of a laser light pulse to determine distance. Versions of the LiDAR methods have been used since the middle of the 20th century, and has for example been popular in altitude measurement applications for aircraft. A 3D LiDAR, see Figure 3.12a, refers to a device that combines multiple distance measurements in different directions, where the output is a point cloud of the surrounding environment. An example of such a point cloud can be seen in Figure 3.12b. 3D LiDARs have gained popularity in recent years, much due to the progress and advancements in autonomous vehicle technology.

The working principle of a LiDAR is that a laser pulse is generated and sent in a certain direction. When that laser pulse encounters an object, it is reflected back towards the receiver in the LiDAR. The time it takes for the light pulse to travel is measured and used for determining the distance to the object, as seen in Figure 3.13.

In this thesis, evaluation and comparison of multiple commercially available 3D LiDARs has been performed. As no 3D LiDAR has been available to perform physical tests on, the analysis has been made mostly on a theoretical basis using the specifications from existing devices.

(a) An example of a 3D LiDAR, Ouster OS1. (b) Visualization of the pointcloud generated from a 3D LiDAR.

Figure 3.12: 3D LiDAR device and pointcloud. [*Ouster OS1 3D LiDAR* 2020]

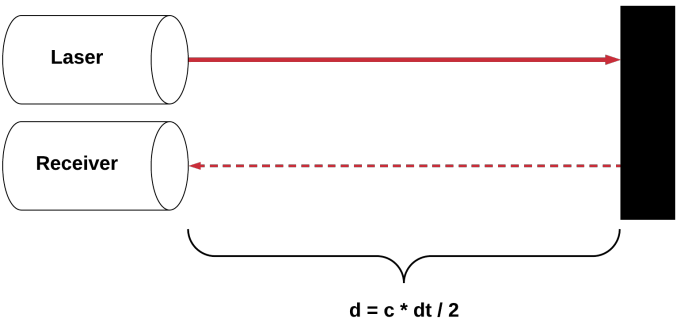


Figure 3.13: Working principle of a LiDAR the distance d can be determined by multiplying the speed of light, c, with the time elapsed, dt, from transmission to reception of the light pulse, and dividing by 2.

When using LiDARs for the purpose of small object detection, a number of requirements exist. Most importantly, the LiDAR needs to be able to detect the MAV at a long enough range to facilitate a control input early enough in the approach phase. A sufficient angular field of view (FoV) in both the horizontal and vertical directions are required. For the horizontal direction, most 3D LiDARs use a 360° FoV, which is more than sufficient for this scenario. For the vertical FoV, there needs to be a large enough interval to fit everything from the water surface and up to the angle at which the approach glideslope takes place, with margin. In addition to that interval, there is potentially a need for a margin to account for changes in the orientation of the boat. This is further elaborated in in the "Implementation" section below. Another

important factor is the angular resolution of the laser measurements. To explore this aspect, LiDARs with different resolutions have been chosen for evaluation. Among state-of-the-art 3D LiDAR producers, Velodyne and Ouster have products that are commonly used and they provide accessible datasheets and specifications. This is why their products have been chosen for the comparison. Three models with appropriate FoV and long specified range have been chosen. The models evaluated are Velodyne Puck, Ouster OS1 and Velodyne Alpha Prime. The most relevant characteristics are summarized in Table 3.4 below.

	Velodyne Alpha Prime	**Velodyne Puck**	**Ouster OS1**
Maximum range	245 m	100 m	120 m
Horizontal FoV	360°	360°	360°
Vertical FoV	40°	30°	45°
Horizontal resolution	0.1-0.4°	0.1-0.4°	0.2-0.8°
Vertical resolution	0.11°	2°	0.7°
Refresh rate	5-20 Hz	5-20 Hz	10-20 Hz
Laser wavelength	903 nm	903 nm	865 nm

Table 3.4: Specifications of LiDAR products. The horizontal resolution and refresh rate can be configured together, to achieve either higher resolution, or higher refresh rate. [*Ouster OS1 Datasheet* 2020] [*Velodyne Product Guide* 2020]

When studying the properties of a 3D LiDAR, there are a few important characteristics to take note of. The range measurement is usually accurate to a few centimeters, depending on the range of the object. Combined with a low angular error (0.01° for the Ouster OS1), the position accuracy and precision of 3D LiDARs can be expected to be enough for the purpose of landing the MAV on the boat. The actual position error depends of how the returned point cloud is processed. This is expanded upon in the "Implementation" section.

Other aspects of the 3D LiDAR can be analyzed by studying the optical echo power, which can be expressed by a range finder equation,

$$P(R) = P_0 \rho \frac{A_0}{\pi R^2} \eta \exp(-2\alpha R) \tag{3.5}$$

where $P(R)$ is the power detected by the LiDAR receiver, R is the distance between the LiDAR and object, P_0 is the power in the emitted light pulse and A_0 is the area of the receiver, η is a coefficient describing power losses due to hardware and geometry configurations in the LiDAR, ρ is a reflectance coefficient related to characteristics in the object, and α is an extinction coefficient that describes the attenuation of the light energy when travelling through air [Wojtanowski et al., 2014]. [Alkholidi and Altowij, 2012]

The most relevant aspect of the equation is that a fraction of the emitted power will be reflected back to the LiDAR. The size of this fraction depends on a number of factors, two of which are relevant to discuss with the target scenario of this thesis in mind. Firstly, the extinction coefficient α, which describes absorption and scattering in the air, is affected by environmental factors such as rain and fog. Secondly, the reflectance ρ varies with the color, material and angle of incidence of the object. In addition, the relationship between the sizes of a beam laser-spot and the detected object is also a significant factor. All of these factors affect the received power, which in turn limits the maximum detection range of objects for the LiDAR.

Extinction

As seen in Equation (3.5), the power attenuation caused by the extinction coefficient α is nonlinear. A doubling of α results in a 63% decrease in received power, which means that the receiver is significantly affected by rain. Therefore it is important to analyze the impact of rain on the maximum detectable range.

The attenuation caused by the extinction can be attributed to two factors, scattering and absorption. The extinction coefficient is the sum of the two. The relationship between scattering coefficient, absorption coefficient and rainfall rate can be seen in Table 3.5 [Guo et al., 2015].

	Scattering coeff.	Absorption coeff.
Light rain (5 mm/hr)	0.0013 m^{-1}	0.00132 m^{-1}
Moderate rain (12.5 mm/hr)	0.0024 m^{-1}	0.00244 m^{-1}
Heavy rain (25 mm/hr)	0.0038 m^{-1}	0.00387 m^{-1}

Table 3.5: Values for the scattering coefficient and absorption coefficient for rainy conditions. This data is based on a laser wavelength slightly larger than the ones used by the LiDAR devices evaluated in this report.

From Table 3.5 we can approximate a proportional relationship between the extinction coefficient and the rainfall rate where we let the

$$\alpha = k_{ex} r \tag{3.6}$$

where $k_{ex} = 0.00044 \frac{hr}{mm_{rain} m}$ and r is the rainfall rate in mm per hour. We can derive the relationship of the rainfall rate and the corresponding maximum range using Equation (3.5), which yields

$$r = -\frac{\ln \frac{PR^2\pi}{P_0 \rho A_0 \eta}}{2k_{ex}R} \tag{3.7}$$

The LiDAR detects an object if a power $P \geq P_{min}$ is returned. We let P_{min} be determined by the device specific maximum range at optimal scattering conditions ($\alpha = 0$), such that $P_{min} = P(R_{max})$, which yields

$$P_{min} = P_0 \rho \frac{A_0}{\pi R_{max}^2} \eta \tag{3.8}$$

With $P = P_{min}$, combining Equation (3.7) and (3.8) yields

$$r = -\frac{\ln \frac{R}{R_{max}}}{k_{ex} R} \tag{3.9}$$

From equation (3.7) the maximum detection range as a function of the rainfall rate can be plotted, as seen in Figure 3.14.

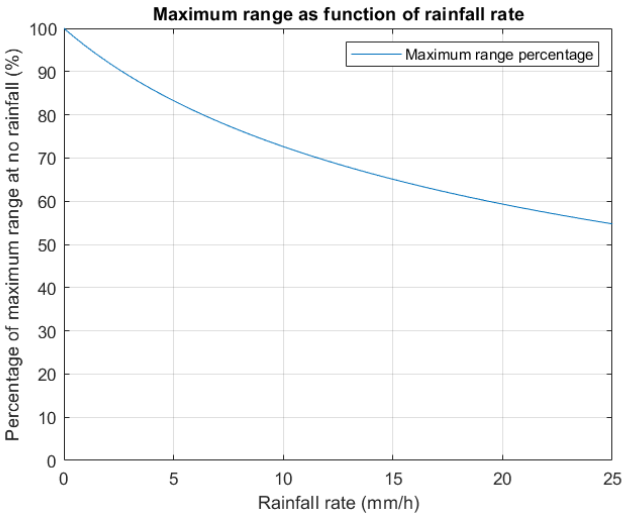


Figure 3.14: The theoretical maximum range of LiDAR devices, expressed as a percentage of its maximum range at no rainfall, as a function of rainfall rate.

Reflectance

Reflectance is an important phenomenon when it comes to LiDAR performance. The reflectance of an object, the ratio of reflected light energy and incident light energy, acts as an attenuation of the light pulse sent from the LiDAR and thus factors into the maximum working range for a LiDAR. In Equation (3.5) the reflectance coefficient is a linear contribution to the power loss. As a reference of the effect of the object reflectance on the maximum distance detectable by LiDAR, we can use equation (3.5), taking a few considerations into account. In [*Ouster OS1 Datasheet*

2020], the maximum range is specified under the condition of a 80% reflectivity, and we assume it is under optimal extinction conditions ($\alpha = 0$). Rearranging equation (3.5)

$$\rho = \frac{P\pi R^2}{P_0 A_0 \eta \exp(-2\alpha R)} \tag{3.10}$$

If we let the minimal detectable power $P_{min} = P(R_{max}, \rho_{max})$ where the $\rho_{max} = 0.8$ we get

$$R = R_{max}\sqrt{\frac{\rho}{\rho_{max}}} \tag{3.11}$$

A plot of the theoretical impact of the reflectance on the maximum range of the LiDAR is shown in Figure 3.15.

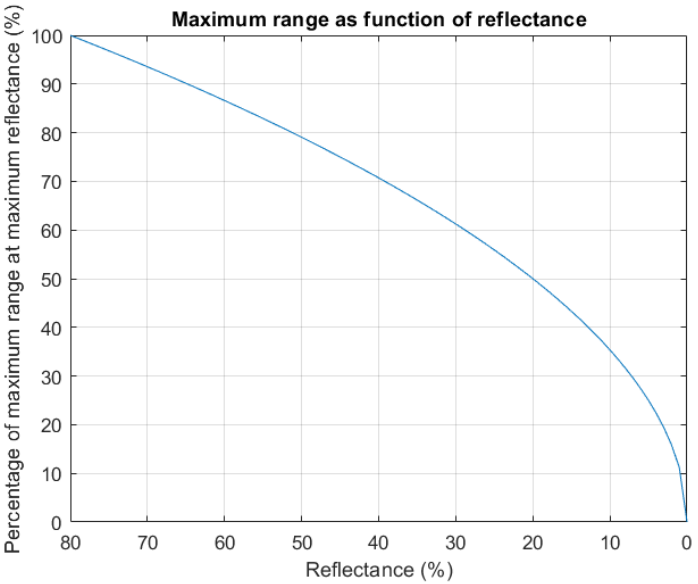


Figure 3.15: The impact of reflectance on the maximum range of a LiDAR. This is based on specifications from the Ouster OS1.

Beam Divergence

The laser beams of LiDARs have a certain divergence. This causes the intensity of the light pulse to spread out over a larger area, the spot size. In the application of detecting small objects at a distance, this has the effect of limiting the reflected power and in turn also the maximum range which an object can be detected at.

For the Ouster OS1, this divergence is specified as 0.18° which means that the laser spot size is as large as the height of the MAV (around 5 cm) at a distance of approximately 16 m.

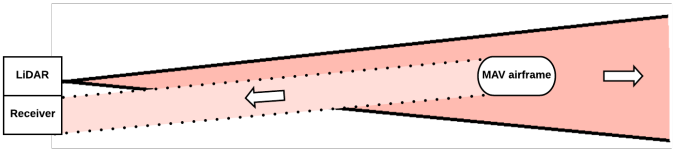


Figure 3.16: The divergence of single laser beams limit the power reflected back to the LiDAR when the object is at a large distance.

Implementation

As the weight of 3D LiDARs usually ranges from hundreds of grams an up, they are most suitable for the "active boat, passive MAV" configuration. A 3D LiDAR needs a clear line of sight to the space behind the boat in which the MAV approaches. One possible way of mounting the device can be seen in Figure 3.17. Since the boat changes orientation depending on the waves and velocity, the orientation of a rigidly mounted 3D LiDAR changes with it. There are two major ways of countering this. In the first way, the relative position between MAV and boat, expressed in the vehicle frame of the boat, can be calculated with the knowledge of the orientation of the boat. This can be achieved with an Inertial Measurement Unit (IMU). A chart of this solution can be seen in Figure 3.18. A problem with this solution is that the boundaries of the FoV change and the MAV have a higher probability of going out of view when the boat encounters large pitch and roll angles. This sets a minimum boundary required for a sufficient FoV. The value of this boundary is a function of how much the boat exerts pitch and roll, the altitude of the MAV when it gets detected by the 3D LiDAR, and at what distance that detection occurs.

Another solution would be to mount the 3D LiDAR on a gimbal that stabilizes its orientation. In this way the relevant relative position can be acquired without any transformation of the sensor reading. A problem with this solution is the addition of an extra component with potential of failure. The gimbal would have to be robust and withstand water and weather.

The actual point cloud returned by the device has to be processed to yield a MAV position. One basic way to do this is to define a volume in which the MAV is expected to approach within, which excludes all points that are false positives and might come from the boat, people in the line of sight or other objects. In this volume of interest, a collection of points close to each other could be identified as the

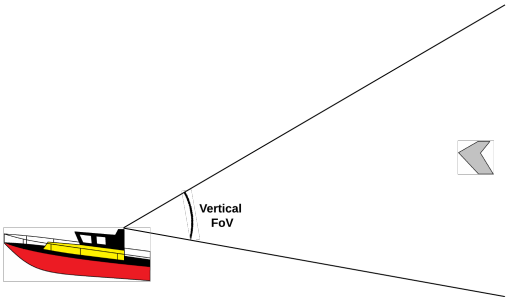


Figure 3.17: 3D LiDAR mounting alternative.

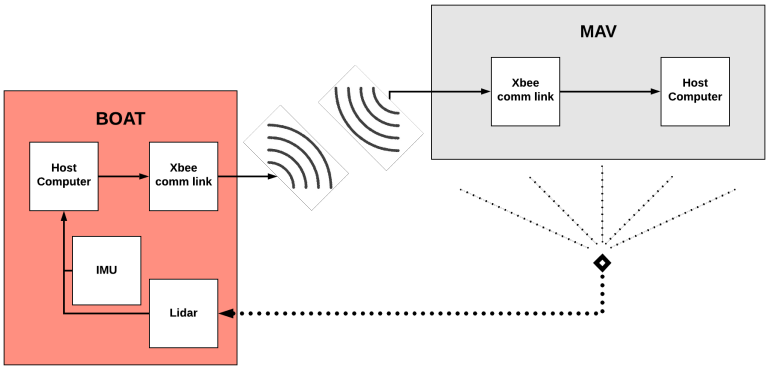


Figure 3.18: Chart of a potential LiDAR implementation.

MAV and the average position of those points could be used as the sensed position. There are several papers written on alternative methods in this area. For example, in [Hammer et al., 2018], 3D object detection algorithms are developed for the purpose of detecting small scale unmanned aerial vehicles. The results of the paper indicate that the algorithms are successful but depend strongly on range and resolution.

Testing

As no physical 3D LiDAR device was available, no actual tests were performed. However, there are several important factors that need to be tested in order to properly evaluate the suitability of such a device in the way described in the "Potential Implementation" section. Presented below are a number of these factors.

The maximum range at which a 3D LiDAR device is able to detect a MAV should be tested. This could be done by hanging the MAV from thin lines and finding

the range. Different orientations of the MAV would have to be considered as the difference in visible area is affected by this.

The degree of the 3D LiDAR being prone to report false positive hits, also referred to as "ghost hits" is important to test because it can make the detection of the MAV hard if ambiguities emerge. The influence of rain on this phenomenon is also important to evaluate as it could limit the utility of this positioning method in rainy conditions.

Modeling

To simulate a 3D LiDAR in the target scenario, all of the laser light beams were modelled as lines from one single point that has a fixed translational and rotational offset from the origin of the boat, seen in the body frame of the boat, aiming rearwards. A visualization of this can be seen in Figure 3.19. The device specific FoV, resolution and refresh rate were used. In the simulation, the MAV was modeled as two rectangular planar areas representing its approximate cross section seen from its front, and from below it. In every sensor reading in the simulation model, all intersections of any of the two MAV planes and a laser beam were recorded as a hit, to which a random Gaussian error was added in the device specific error range. Since the error information only contained the magnitude, it is assumed that a Gaussian distribution is appropriate. For every LiDAR reading, all hits were stored in an array from which the average position was returned and used as the scanned position.

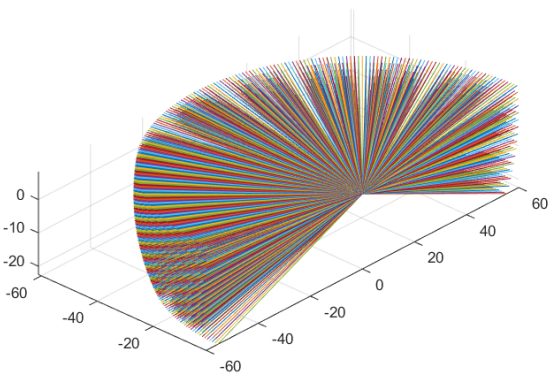


Figure 3.19: 3D LiDAR laser beams visualized as lines.

A complex value to estimate is the actual maximum LiDAR range at which a de-

tection of a MAV can take place. As mentioned, this depends on many factors and cannot be verified without a physical 3D LiDAR device. To account for this in the simulation model, two values were used to limit maximum detection range, 60 m and 30 m.

3.7 Instrument Landing System

In commercial aviation, the Instrument Landing System (ILS) is the most commonly used guidance system for aircraft in the approach and final stages of the flight. It was first introduced in 1964 and is now present at the majority of large airports. The system allows pilots or autopilots to navigate through the approach stage of the flight regardless of sight conditions.

ILS works by transmitting radio waves with different properties to the left side and right side of the center line of the runway intended to be landed on. This is achieved by using phased antenna arrays positioned on the far end of the runway, as seen in Figure 3.20 and 3.21. These signals are received and processed into an angular deviation from the predetermined approach path in the receiving aircraft. This allows the aircraft to control course and pitch to align with the approach path and line up correctly with the runway. The system mainly comprises two subsystems referred to as the localizer and the glideslope. The localizer is responsible for the horizontal guidance and runway alignment of the aircraft. The glideslope is responsible for the vertical guidance. [Van Valkenburg, 2002] [*Ground-Based Navigation - Instrument Landing System (ILS)* 2020]

The actual positioning principle of ILS is based mainly on two techniques, one referred to as transmitter modulation and the other as space modulation. In the transmitter modulation process, a carrier sine wave in the Very High Frequency (VHF) range is generated and split up into two separate carrier waves with the same frequency, phase and amplitude. Each one is sent to a radio modulator unit which modulates a 90 Hz or 150 Hz sine wave onto the carriers. In the space modulation process, the phase of the 90 and 150 Hz waves are altered and the two signals are combined, before being distributed to different sections of the antenna array. Through directional radiation of the antenna array, the 150 Hz modulated signal dominate the right side of the runway, seen from an approaching airplane, and the 90 Hz modulated wave dominate the left side. The resulting configuration creates an interference pattern that alters the depth of modulation of the received signals depending on the angular offset of the receiving airplane, the principle of which can be seen in Figure 3.22. The difference in depth of modulation between the 90 Hz dominant wave and the 150 Hz dominant wave is what is measured and used for positioning in the aircraft. [McCollum, 1983]

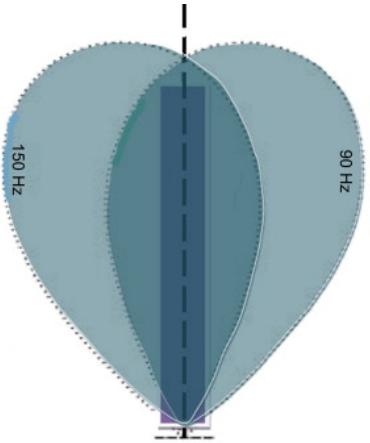


Figure 3.20: The ILS glideslope and localizer antenna patterns. The signals with different frequencies are radiated in different directions which makes it possible to determine position relative to the approach path. [Acharya, 2014a]

Figure 3.21: ILS antenna array consisting of multiple Log-Periodic Dipole Array (LPDA) antennas. [*VHF Nav* 2020]

Implementation

With inspiration from the ILS positioning system, there are multiple ways to potentially implement an approach path following system. As the full extent of the ILS system is immensely complex and comprehensive, this thesis focuses on a significantly scaled down version. In order to evaluate what can be achieved using relatively simple and accessible components, this thesis explores a version using directional antennas and RSSI measurements. A possible way of achieving a guidance path for the MAV to follow could be to mount two directional antennas on the boat as seen in Figure 3.23. This way, two independent radio signals can be sent out and the difference between the signal strength of the two can be used as a measurement of the approach path deviation.

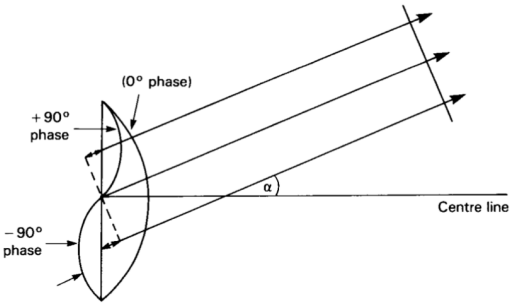


Figure 3.22: An angular offset from the center line of the runway is detected through a phase difference that is measurable on the aircraft. [Smith, 1988]

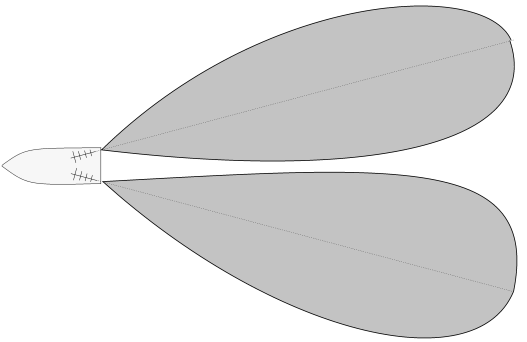


Figure 3.23: Possible configuration for mounting of antennas and corresponding main lobe radiation patterns.

Directional antennas come in multiple versions such as parabolic antennas, panel antennas, Log-Periodic Dipole Array (LPDA) or Yagi antennas. They have radiation patterns that make the signal strength vary with the angle offset from their center beams. A directional antenna is usually defined by its antenna gain, a number that reflects how much it is able to direct its radiation in a specific direction, measured in dBi (Decibel isotropic). The unit is defined as the ratio of an antenna's signal strength at its center beam, and the signal strength of ideal antenna with a completely spherical, isotropic radiation pattern. Another defining property of a directional antenna is its beam width, which specifies the width of the radiation beam. More specifically, it is defined as the angle of the circular sector of its radiation pattern, that has at least -3 dB radiated power compared to the power in its center beam.

There are a number of limitations and requirements to take into account when discussing radio frequency methods. Firstly, the available and license free radio

frequencies are limited to the Industrial, Scientific and Medical (ISM) bands which limits the number of available frequencies and the amount of power that can be used. Secondly, directional antennas often have a smaller beam width and more directionality for shorter wavelengths, i.e. higher frequencies. For the purpose of the guidance system, it would be required to have a fairly small beam width because of the need of a relatively large change in signal strength for a certain deviation in flight path. This in turn requires relatively high frequencies.

Testing

Setup To evaluate how well the directional antenna concept works, a number of tests were performed. A directional antenna was set up and the received signal strength was measured at different angles on an open field. The antenna used was a panel antenna with a beam width of 60° and an antenna gain of 9 dBi. In order to measure the signal strength, a pair of the Xbee communication devices described in the "Communication" section were used. The devices have a Received Signal Strength Indication (RSSI) measurement available. One of them was connected to the directional antenna and used as the transmitter, The other was used as an RSSI measurement device. The setup can be seen in Figure 3.24.

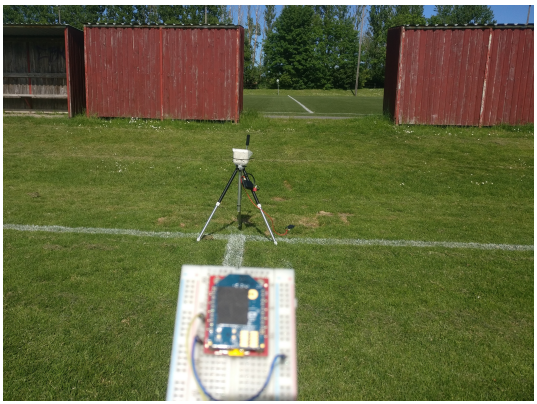


Figure 3.24: Test setup with a directional antenna and an Xbee device.

First, the Xbee transmitter was set to a low power mode in order to keep the receiver from saturating the RSSI measurement which occurred at close distances and close to the center beam. The signal was then recorded at the center beam of the antenna to measure the RSSI value and the noise of the signal. The measurement can be seen in Figure 3.25.

To measure how the measured RSSI value changed with the angle offset from the directional antenna, measurements were taken every 5° starting from the center beam and up to 90°, at a constant distance of about 20 m away from the directional antenna. The same test was done three times to evaluate repeatability.

Results The result of the noise test can be seen in Figure 3.25.

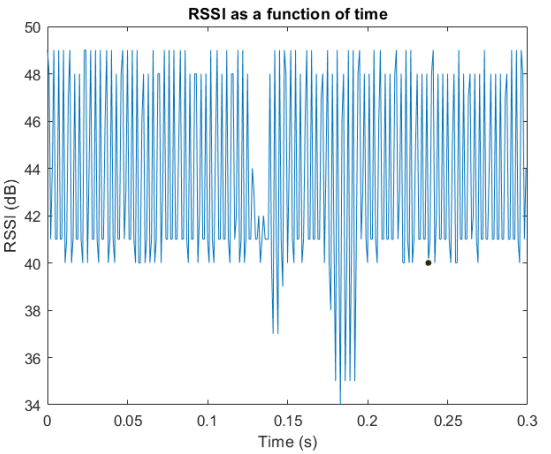


Figure 3.25: RSSI plot during a short time period, at the center beam of the directional antenna.

The result of the test of RSSI value as a function of angle offset from the center beam can be seen in Figure 3.26.

Discussion The result of the first test shows that there is significant noise in the RSSI measurement. When combined with the result of the second test, it can be seen that the noise is almost 15 dB, and the signal spans approximately 18 dB which yields a signal-to-noise ratio of slightly over 1. The measuring method supported in the Xbee module consisted of a Pulse Width Modulated (PWM) signal which had to be read from a micro controller. As a result, the micro controller returned the measured value in steps of 1 dB, which is not precise. Further, when measuring the RSSI at every 5°, there was a significant inconsistency, causing the whole "envelope" to jump up or down for a short moment, making the measurement less precise. To summarize the test, there was a large amount of troubling factors and the precision of the measurement cannot be expected to be enough to be used for this type of navigation.

The antenna used for the test had roughly the double beam width than the type that would be desirable for guidance purposes. An antenna with a smaller beam width would have had a larger antenna gain and in turn a larger signal to noise ratio

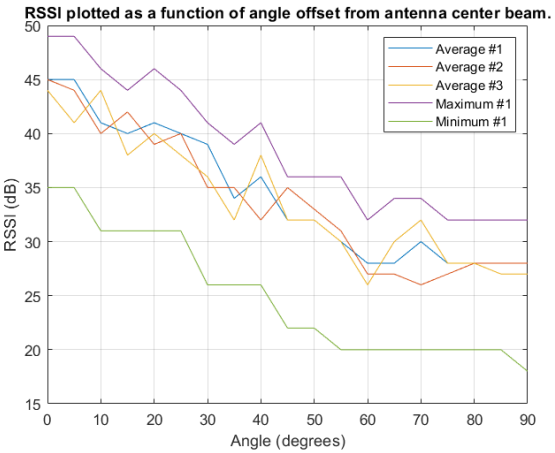


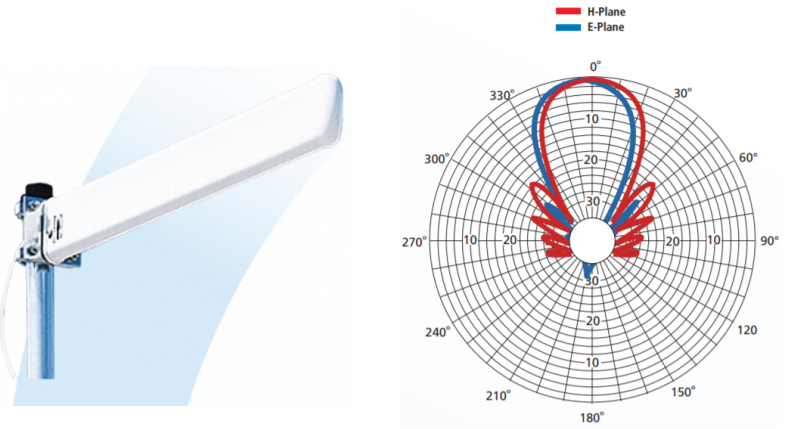
Figure 3.26: Measured RSSI as a function of the offset angle from the antenna center beam.

could have been achieved. However, it is not likely that it would have increased the quality of the signal enough to make it a functioning guidance system.

As only one pair of Xbee devices were available, this test only takes into account one of two signals required for guidance. It is not clear how much the two signals would affect each other and there is a risk that the measurement would have been worse with two parallel systems.

Modeling

To model the small scale ILS system, a few subsystem models were created. The system was modeled such that two directional antennas were positioned on the boat with a certain directional angular offset according to Figure 3.23. The antenna model was based on a commercially available Yagi antenna, Laird Connectivity PC2415N, which has a beam width of around 30 degrees. This model can be seen in Figure 3.27.



(a) The Laird PC2415N Yagi antenna.

(b) The radiation pattern of the The Laird PC2415N Yagi antenna.

Figure 3.27: The evaluated antenna and its radiation pattern. [*Laird PC2415N datasheet* 2020]

First, the RSSI value was calculated using a completely spherical radiation pattern such that the distance between MAV and antenna determined the magnitude. The value was then multiplied with the antenna gain, and then compensated for the offset angle of the MAV from the antenna center beam, according to the radiation pattern shown in Figure 3.27b. However, after the RSSI and directional antenna tests were performed and the results showed the disadvantages of this method, the modeling and simulation of this concept was down prioritized and no further analysis was made based on this method.

3.8 Summary

In summary, three positioning systems were investigated; relative GNSS, 3D LiDAR and a small scale version of ILS. Based on testing, GNSS showed promising results and is evaluated further in simulation. 3D LiDAR was investigated on a theoretical basis and a model was set up to be investigated through simulations. The proposed small scale ILS method was experimentally tested, but the results indicated flaws in precision. The method was not investigated further.

4

Autonomous Landing

In this chapter the landing strategy used for landing of the MAV on the boat will be explored. To facilitate this discussion, the chapter details control theoretical tools, design logic and practical considerations for a real-time implementation on a MAV.

The chapter starts with a short look at the PX4 flight controller stack in order to design a strategy compatible with its current functionalities. After this the low level controllers are addressed, proceeding with a discussion on the high-level controllers with a detailed look on the MPC used for generation of feasible flight trajectories, both in a linear and nonlinear setting. This is followed by a discussion on additional decision logic and strategies used for the landing. Finally, a section on system identification for the parameters of the internal MAV-model used by the Model Predictive Controller is included.

A few things such as aborting or retrying a landing will not be considered, i.e. a landing will always push trough even if it might be obvious and detectable that there is no chance of successfully completing the landing. In Figure 4.1 below, an overview of the modules involved in the landing are included to make it easier following the discussion and getting a quick overview on how the modules are relating to each other.

4.1 PX4 Flight Controller

PX4 is an open source autopilot software mainly used for flying drones such quadcopters and fixed-wing airplanes. There are also a number of available hardware platforms compatible with the software stack [*PX4 - Autopilot User Guide* 2020].

The flying wings currently used by SSRS are using the PX4 flight stack (or Ardupilot, a autopilot software similar to PX4) for their missions. Thus, the proposed landing strategies focus mainly on the high level control and planning of the landing

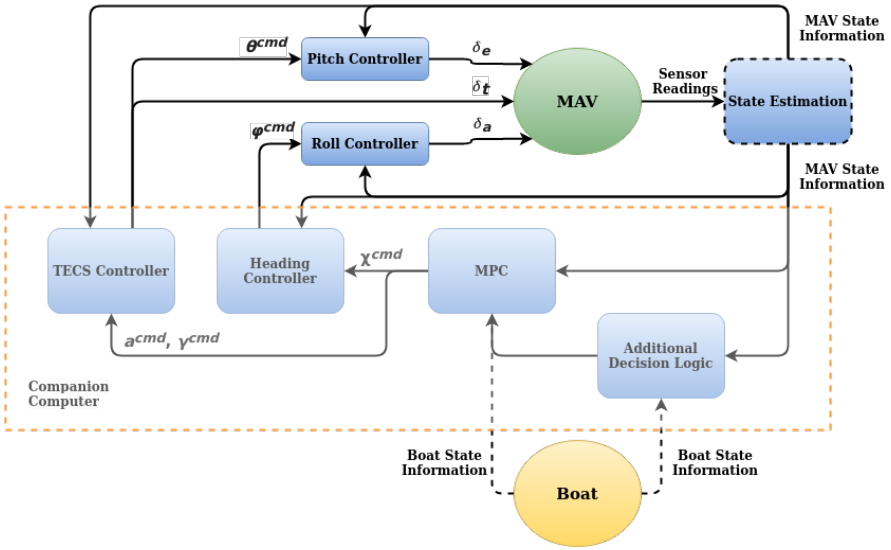


Figure 4.1: Overview of the control modules used for the autonomous landing. The modules intended for implementation in the companion computer is marked within the orange box. The state estimation was not implemented in the simulation. Although, the state estimation module is directly available in the PX4 flight stack.

rather than low level control and state estimation, already fully functioning and implemented in the PX4 flight controller. Since the PX4 software is running on a microcontroller, it is well suited for reading sensors, running state estimators, stabilize the flight and control the vehicle to execute commands and missions. If more computationally demanding tasks need to be performed or if external hardware, not compatible with PX4 is to be used, a companion computer can be attached via the MAVLink communication protocol. MAVLink is a lightweight communication protocol made for drones, and it can be used to communicate with the PX4 system by reading sensors, logging system status or uploading new missions and instructions. In the so called *offboard mode*, a companion computer can continuously control the actions of the PX4 controlled vehicle, for example by setting attitude, angular rates or way-points to follow. The MAVLink protocol can also interface with the Robot Operating System (ROS) via the MAVROS node. The proposed controllers were designed with this system architecture in mind. As long as a controller or planner can be implemented in ROS (Python or C++) it can run on a companion computer and simply send commands to the PX4 in offboard mode via MAVROS and MAVLink.

4.2 Low Level Control

This section will focus on how to perform the low level control needed for the landing strategies used. All controllers that are generic for a flying wing, and thus not directly related to the landing sequence, will be considered low level controllers. In this report, a general characteristic for the low level controllers is that the information used has nothing to do with, and is not in relation to the boat. Hence, the sensors used for these controllers is only telling something about the state of the MAV, not about the boat.

Using the PX4 stack, controllers for pitch and roll angles can be assumed to be accessible and working. Although, for the simulation (see Chapter 5) these controllers have to be implemented and will therefore be described briefly as well. Direct yaw control will not be considered since a flying wing does not have any control surfaces dedicated mainly for yaw control. Instead, a change in yaw is acquired by rolling the plane resulting in a yaw rate.

Pitch and Roll Control

Both roll and pitch are controlled using simple PID and PD-controllers, using the control surfaces (see Section 2.2) of the MAV as seen in Equations (4.1) and (4.2) below

$$\delta_a(t) = k_{p_\phi}(\phi^{cmd}(t) - \phi(t)) + k_{i_\phi}\int_0^t (\phi^{cmd}(\tau) - \phi(\tau))d\tau + k_{d_\phi}p(t) \tag{4.1}$$

$$\delta_e(t) = k_{p_\theta}(\theta^{cmd}(t) - \theta(t)) + k_{d_\theta}q(t) \tag{4.2}$$

where p and q are the body roll rates around the $\mathbf{i}_x^{\mathcal{B}}$ and $\mathbf{j}_y^{\mathcal{B}}$ respectively, used instead of the true derivatives $\dot{\phi}$ and $\dot{\theta}$. A more detailed explanation on successive loop-closing for controlling pitch and roll for a MAV, as well strategies for choosing initial controller gains can be found in [Beard and McLain, 2012]. The parameters used for the simulations were found by manually tuning the initial parameters, and can be seen in Tables B.1 and B.2 in Appendix B.

Heading Control

Since a flying wing aircraft does not have a rudder it can not control the yaw (ψ) or course (χ) angles directly. However, the yaw-rate ($\dot{\psi}$) of the MAV is also coupled with roll, and thus the heading of the aircraft can be controlled by cascading a PI controller with the roll angle controller as in [Beard and McLain, 2012], here written as

$$\phi^{cmd}(t) = k_{p_\chi}(\chi^{cmd}(t) - \chi(t)) + k_{i_\chi}\int_0^t (\chi^{cmd}(\tau) - \chi(\tau))d\tau \tag{4.3}$$

The parameters used for the simulations were found by manually tuning the initial parameters, and can be seen in Table B.3 in Appendix B.

Total Energy Control System

Some control strategies used for longitudinal control of fixed-wing aircraft assume independence between airspeed and altitude. This can lead to undesired effects when coupling is active [Argyle and Beard, 2016]. An alternative is to use a total energy control system (TECS), a control system for longitudinal control which makes use of the altitude-airspeed coupling. Instead of separately controlling airspeed and altitude the idea is to simultaneously control the *total energy* of the system and the *energy balance* between potential and kinetic energy. The total energy is controlled by the throttle, adding energy to the system, and by the drag force, continuously removing energy from the system. The energy balance is controlled by sending commands to the pitch-controller, which in turn is making a conversion between airspeed and altitude. For this project two different versions of TECS were used, one controlling energies and the other one controlling energy rates.

Energy Controller Using this type of controller, the quantities desired to control are altitude (h) and airspeed (V_a), which can be done indirectly by simultaneously controlling the total energy and the energy balance. The total energy of a the MAV can be expressed as the sum of kinetic and potential energy as following

$$E_T = U + K = mgh + \frac{1}{2}mV_a^2 \tag{4.4}$$

where U is the potential energy, K is the kinetic energy and h is the altitude. In a similar way the energy balance between potential and kinetic energy can be expressed as

$$E_B = U - K = mgh - \frac{1}{2}mV_a^2 \tag{4.5}$$

With the desired setpoints for velocity and altitude as V_a^{cmd} and h^{cmd}, two PI controllers can be designed independently to minimize the total energy and energy balance error respectively.

$$\delta_t = k_{p,\delta_t}(E_T^{cmd} - E_T) + k_{i,\delta_t}\int_0^t (E_T^{cmd} - E_T)d\tau \tag{4.6}$$

$$\theta^{cmd} = k_{p,\delta_e}(E_B^{cmd} - E_B) + k_{i,\delta_e}\int_0^t (E_B^{cmd} - E_B)d\tau \tag{4.7}$$

where δ_t is the motor control command and θ^{cmd} is the command sent to the low level pitch PD-controller [*Total Energy Control for Longitudinal Autopilot* 2020].

Energy Rate Controller In a similar way, a controller can be designed based on energy rates. The quantities desired to control are now flight path angle (γ) and acceleration $\dot{V}_a$, which can be done indirectly by simultaneously controlling total energy rate and energy balance rate. Time derivatives of equation (4.4) and (4.5) yields

$$\dot{E}_T = mg\dot{h} + mV_a\dot{V}_a \tag{4.8}$$

$$\dot{E}_B = mg\dot{h} - mV_a\dot{V}_a \tag{4.9}$$

Usually scaled versions of (4.8) and (4.9) are used [Argyle and Beard, 2016]. These scaled versions are denoted as $\dot{E}_t$ and $\dot{E}_b$, and are defined below.

$$\dot{E}_t = \frac{\dot{E}_T}{mgV_a} = \frac{\dot{V}_a}{g} + \frac{\dot{h}}{V_a} \tag{4.10}$$

Using $\gamma = tan(\frac{\dot{h}}{V_a})$, a small angle approximation yields

$$\dot{E}_t = \frac{\dot{V}_a}{g} + \gamma \tag{4.11}$$

Similarly for the energy balance rate

$$\dot{E}_b = \frac{\dot{E}_B}{mgV_a} = -\frac{\dot{V}_a}{g} + \frac{\dot{h}}{V_a} = -\frac{\dot{V}_a}{g} + \gamma \tag{4.12}$$

Corresponding PI controllers for $\dot{E}_t$ and $\dot{E}_b$ can now be designed using desired $\dot{V}_a^{cmd}$ (equal to a^{cmd} if the wind is constant in the direction of flight) and γ^{cmd} as inputs to the TECS energy rate controller.

$$\delta_t(t) = k_{p,\delta_t}(\dot{E}_t^{cmd}(t) - \dot{E}_t(t)) + k_{i,\delta_t}\int_0^t (\dot{E}_t^{cmd}(\tau) - \dot{E}_t(\tau))d\tau \tag{4.13}$$

$$\theta^{cmd}(t) = k_{p,\delta_e}(\dot{E}_b^{cmd}(t) - \dot{E}_b(t)) + k_{i,\delta_e}\int_0^t (\dot{E}_b^{cmd}(\tau) - \dot{E}_b(\tau))d\tau \tag{4.14}$$

A more detailed explanation and more advanced variants of the TECS controller can be found in [Argyle and Beard, 2016]. The parameters used for the simulations was found by manual tuning, and can be seen in Table B.4 in Appendix B.

4.3 High Level Control

To perform an autonomous landing, decisions on desired direction of flight continuously have to be made. An approach could be to make a high level decision once and then follow the generated reference trajectory as in [Fridén, 2020]. This approach could work well for a landing on a static target but is unlikely to be suitable for a scenario with a dynamic target, where and position and velocity is changing over time. For a problem quite similar to the target scenario, a strategy based on PID controllers was tested in [Persson, 2016], but the performance of the landing was highly dependant on the starting conditions of the landing being correct. Continuation of this work indicates that a strategy based on online optimization is performing better than the initial PID approach [Persson, 2019].

In our case a PID approach could be achieved by simply cascading one more layer of PID controllers with the existing low level controllers described above. Here the left/right motion would be controlled with the heading controller and in the up/down motion by sending altitude commands to the energy version of the TECS. However, previous studies show that the online optimization strategy seems more promising than a PID approach. Also, the proposed ILS positioning system, suitable to combine with a PID landing strategy, seems unfit for the purpose due to the hardware constraints outlined in Section 3.7. Thus, focus was shifted towards the optimization based strategy Model Predictive Control (MPC).

4.4 Model Predictive Control

The first part of this section will present the fundamentals of model predictive control (MPC) and the later part will cover how this can be used to plan an execute an autonomous landing. Practical considerations for real time implementation are also discussed towards the end of the section.

Introduction

Model predictive control (MPC) is an online optimization based control strategy. The main idea of MPC is to continuously optimize the plant behaviour over a predefined number of discrete steps into the future, in a receding horizon fashion. A MPC is relying on the availability of a discrete time model of the plant being controlled. The plant model is used to give the MPC an approximate idea of the plant dynamics, which can be used to predict how future states of the plant will propagates given certain control inputs. Using this knowledge of plant model dynamics, as well as other system constraints such as physical actuator limits, the MPC is running optimizations in fixed time intervals. In each optimization the plant model behaviour is optimized over a fixed number of steps into the future by finding the corresponding sequence of control inputs. The first control input on the horizon is

then used as a control command to the real plant. The rest of the sequence is discarded as new sequence is given in the optimization at the next time-interval. The number of time-steps which the plant model behaviour is optimized over in each iteration, is called the *prediction horizon* (H_p). The corresponding number of control inputs is called the *control horizon* (H_u). The principle described is visualized in Figure 4.2. The control horizon is naturally always shorter than the prediction horizon. If $H_u = H_p - 1$ there are one control input for each step along the prediction horizon. If $H_u < H_p - 1$ the control input is usually held constant for the last steps along H_p.

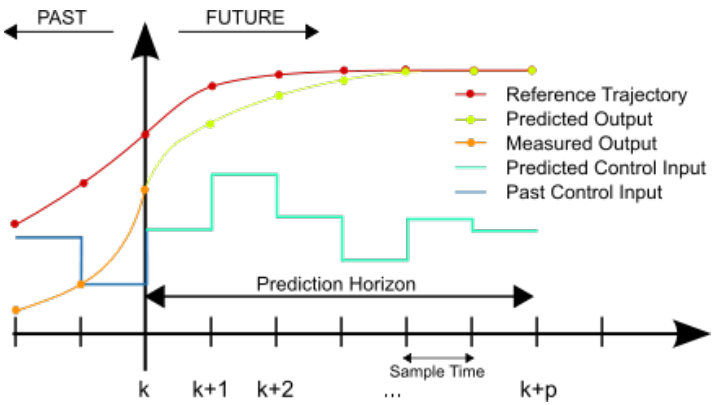


Figure 4.2: Visualization of the working principle of a MPC at time k. With the notation used in the report p in the figure is corresponding to $H_p - 1$. [Wikimedia Commons, 2007]

Quadratic Programming

The optimization is a central part of the MPC. The focus of optimization in this report will be on problems that can be formulated as a quadratic program (QP), as this type of problem is well documented and can be solved efficiently (more details on the speed for solving a QP is given later in Section 4.4). A QP consists of a quadratic objective function that is to be minimized (or maximized), possibly subject to a set of linear inequality constraints and/or a set of linear equality constrains

$$\begin{aligned} J = \min_{\bar{\mathbf{x}}} \quad & \frac{1}{2}\bar{\mathbf{x}}^T \bar{H} \bar{\mathbf{x}} + \bar{\mathbf{f}}^T \bar{\mathbf{x}} \\ \text{subject to} \quad & \\ & \bar{F}_{ineq}\bar{\mathbf{x}} \leq \bar{\mathbf{b}}_{ineq} \\ & \bar{F}_{eq}\bar{\mathbf{x}} = \bar{\mathbf{b}}_{eq} \end{aligned} \tag{4.15}$$

where $\bar{H}$, $\bar{F}_{ineq}$ and $\bar{F}_{eq}$ are matrices, and $\bar{\mathbf{f}}$, $\bar{\mathbf{b}}_{ineq}$ and $\bar{\mathbf{b}}_{eq}$ are column vectors. In order to use a standard QP-solver, an optimization problem should be formulated accordingly.

Linear MPC Formulation

One system that can be expressed as QP is a linear MPC. For a discrete linear system at time k, with state vector $\mathbf{x}(k) \in \mathbb{R}^{n\times 1}$ and control vector $\mathbf{u}(k) \in \mathbb{R}^{m\times 1}$

$$\mathbf{x}(k+1) = A\mathbf{x}(k) + B\mathbf{u}(k) \tag{4.16}$$

an initial basic MPC formulation can be made. Defining the current state of the plant as $\mathbf{x}_0$, aiming to follow a reference trajectory $\mathbf{x}_r$, while using linear constraints and a quadratic objective function, the optimization at time k can be formulated as follows

$$\min_{\mathbf{u}(k+i)} \sum_{i=0}^{H_p-1} \|\mathbf{x}_r(k+i) - \mathbf{x}(k+i)\|_Q + \sum_{i=0}^{H_u-1} \|\mathbf{u}_r(k+i) - \mathbf{u}(k+i)\|_R$$

subject to

$$\begin{aligned} \mathbf{x}(k+i+1) &= A\mathbf{x}(k+i) + B\mathbf{u}(k+i), \quad i = 0 \ldots H_u - 1 \\ F_{x+i}\mathbf{x}(k) &\le \mathbf{b}_x, \quad i = 0 \ldots H_p - 1 \\ F_{u+i}\mathbf{u}(k) &\le \mathbf{b}_u, \quad i = 0 \ldots H_u - 1 \\ \mathbf{x}(k) &= \mathbf{x}_0, \end{aligned} \tag{4.17}$$

where F_x and F_u are matrices, and $\mathbf{b}_x$ and $\mathbf{b}_u$ are column vectors, together defining the inequality constraints of the optimization. Here, the equality constraints are ensuring that the system dynamics are not violated during the optimization, and the inequality constraints defines allowed boundaries for the solution which for example could be used to set limits on $\mathbf{x}$ and $\mathbf{u}$. The matrices $Q \in \mathbb{R}^{n\times n}$ and $R \in \mathbb{R}^{m\times m}$ are defining the penalties for the state error and control command usage.

A linear MPC like this can be reformulated as Equation (4.15), and thus be solved with a standard QP solver. In order to do this a new vector $\bar{\mathbf{x}}(k)$, stacking all states and control commands along their horizons at time k, is introduced

$$\bar{\mathbf{x}}(k) = [\mathbf{x}^T(k) \ \cdots \ \mathbf{x}^T(k+H_p-1), \ \mathbf{u}^T(k) \ \cdots \ \mathbf{u}^T(k+H_u-1)]^T \in \mathbb{R}^{(nH_p+mH_u)\times 1} \tag{4.18}$$

Similarly for the for the reference trajectory

$$\bar{\mathbf{x}}_r(k) = [\mathbf{x}_r^T(k) \ \cdots \ \mathbf{x}_r^T(k+H_p-1), \ \mathbf{u}_r^T(k) \ \cdots \ \mathbf{u}_r^T(k+H_u-1)]^T \in \mathbb{R}^{(nH_p+mH_u)\times 1} \tag{4.19}$$

The objective function and the constrains in Equation (4.17) can now be expressed as a function of $\bar{\mathbf{x}}(k)$ and $\bar{\mathbf{x}}_r(k)$. Introducing H as a block-matrix built from multiple instances of Q and R

$$H = \begin{pmatrix} Q & 0 & \cdots & 0 & 0 \\ 0 & Q & & & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & & & R & 0 \\ 0 & 0 & \cdots & 0 & R \end{pmatrix} \in \mathbb{R}^{(nH_p+mH_u)\times(nH_p+mH_u)} \tag{4.20}$$

the objective function to be minimized in Equation (4.17) can be expressed as

$$\begin{aligned} J &= [\bar{\mathbf{x}}_r(k) - \bar{\mathbf{x}}(k)]^T H [\bar{\mathbf{x}}_r(k) - \bar{\mathbf{x}}(k)] \\ &= \bar{\mathbf{x}}^T(k) H \bar{\mathbf{x}}(k) - 2\bar{\mathbf{x}}_r^T(k) H \bar{\mathbf{x}}(k) + \bar{\mathbf{x}}_r^T(k) H \bar{\mathbf{x}}_r(k) \\ &= \frac{1}{2}\bar{\mathbf{x}}^T(k)\bar{H}\bar{\mathbf{x}}(k) + \bar{\mathbf{f}}^T\bar{\mathbf{x}}(k) \end{aligned} \tag{4.21}$$

arriving at the desired structure of the objective function. Here the constant term is dropped and

$$\bar{H} = 2H \tag{4.22}$$

$$\bar{\mathbf{f}} = -\bar{H}^T \bar{\mathbf{x}}_r(k) \tag{4.23}$$

Notice that the diagonal structure of H is a result of penalizing states and control signals independently from each other.

In order to satisfy the system dynamics, the equality constraints and initial state from Equation (4.17) should be fulfilled while being expressed as

$$\bar{F}_{eq}\bar{\mathbf{x}} = \bar{\mathbf{b}}_{eq} \tag{4.24}$$

which can be done by defining $\bar{F}_{eq}$ from the matrices A and B (see Equation (4.17)) as

$$\bar{F}_{eq} = \begin{pmatrix} I & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ A & -I & 0 & \cdots & 0 & 0 & B & 0 & \cdots & 0 & 0 \\ 0 & A & -I & & 0 & 0 & 0 & B & & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & & A & -I & 0 & 0 & 0 & & B & 0 \\ 0 & 0 & \cdots & 0 & A & -I & 0 & 0 & \cdots & 0 & B \end{pmatrix} \tag{4.25}$$

and defining $\bar{\mathbf{b}}_{eq}$ as

$$\bar{\mathbf{b}}_{eq} = [\mathbf{x}_0^T \quad \mathbf{0} \quad \mathbf{0} \quad \cdots \quad \mathbf{0} \quad \mathbf{0}]^T \tag{4.26}$$

In a similar way $\bar{F}_{ineq}$ and $\bar{\mathbf{b}}_{ineq}$ can be designed to fulfill the inequality constraints of choice. A common choice is to separately set upper and lower bounds on the states and the control signals as follows

$$\mathbf{x}_{min}(k) \leq \mathbf{x}(k) \leq \mathbf{x}_{max}(k) \tag{4.27}$$

$$\mathbf{u}_{min}(k) \leq \mathbf{u}(k) \leq \mathbf{u}_{max}(k) \tag{4.28}$$

This can be done by defining $\bar{F}_{ineq}$ and $\bar{\mathbf{b}}_{ineq}$ as

$$\bar{F}_{ineq} = \begin{pmatrix} I \\ -I \end{pmatrix} \tag{4.29}$$

$$\bar{\mathbf{b}}_{ineq} = [\bar{\mathbf{x}}_{max}^T(k), \ -\bar{\mathbf{x}}_{min}^T(k)]^T \in \mathbb{R}^{(2nH_p+2mH_u)\times 1} \tag{4.30}$$

where

$$\bar{\mathbf{x}}_{max}^T(k) = [\mathbf{x}_{max}^T(k) \ \cdots \ \mathbf{x}_{max}^T(k+H_p-1), \ \mathbf{u}_{max}^T(k) \ \cdots \ \mathbf{u}_{max}^T(k+H_u-1)]^T \tag{4.31}$$

$$\bar{\mathbf{x}}_{min}^T(k) = [\mathbf{x}_{min}^T(k) \ \cdots \ \mathbf{x}_{min}^T(k+H_p-1), \ \mathbf{u}_{min}^T(k) \ \cdots \ \mathbf{u}_{min}^T(k+H_u-1)]^T \tag{4.32}$$

Notice that the bounds can be set differently for each time-step along the horizon, and that additional inequality-constraints can be added by increasing the number of rows in the $\bar{F}_{ineq}$ matrix and the size of $\bar{\mathbf{b}}_{ineq}$ correspondingly.

Several extensions to the basic linear MPC formulation can be made. One example of a regularly used extension is adding a separate terminal state penalty (i.e a separate Q for $\mathbf{x}(k+H_p-1)$). Another common extension is adding slack variables, giving the opportunity to set soft inequality-constraints (i.e. violations of the constraints are allowed, but are adding a penalty to the objective function).

Nonlinear MPC Formulation

The simplified MAV model, introduced in Section 2.2, does not fit into the linear MPC formulation as it is a continuous system with nonlinear dynamics. In order to use non-linear dynamics in an MPC, the model has to be discretized and the concept of linear MPC has to be expanded further. With a discretized version of the simplified MAV model (f_d), an extension to a Nonlinear Model Predictive Controller (NMPC), allowing for nonlinear objective function, dynamics and constrains, results in the following formulation

$$\min_{\mathbf{u}(k+i)} \quad \sum_{i=0}^{H_u-1} l(\mathbf{x}(k+i),\mathbf{u}(k+i)) + l_T(\mathbf{x}(k+H_p-1))$$

subject to

$$\begin{aligned} \mathbf{x}(k+i+1) &= f_d(\mathbf{x}(k+i),\mathbf{u}(k+i)), \quad i = 0 \ldots H_u-1 \\ f_x(\mathbf{x}(k+i)) &\leq 0, \quad i = 0 \ldots H_p-1 \\ f_u(\mathbf{u}(k+i)) &\leq 0, \quad i = 0 \ldots H_u-1 \\ \mathbf{x}(k) &= \mathbf{x}_0, \end{aligned} \tag{4.33}$$

A nonlinear optimization problem like this cannot be directly solved with a QP-solver. Instead a general purpose nonlinear solver could be used, but this will typically result in longer computational times. Although, if the objective function is kept quadratic the nonlinear optimization problem in Equation (4.33) can be reformulated into a set of optimization problems similar to the one of the linear MPC, where each one can be solved with a QP-solver. One way of doing this is with *sequential quadratic programming* (SQP).

Sequential Quadratic Programming SQP is an iterative method of solving a constrained nonlinear optimization problems. The basic idea of SQP is to use the solution from the previous iteration as the point around which the nonlinear problem is approximated as a QP. In order for the nonlinear problem to become a QP, the objective function is approximated as a quadratic function and the constraints are approximated as linear functions, similarly as in Equation (4.15). This procedure is usually repeated for a fixed number of iterations or until some other terminal condition is met. With an infinite number of iterations the solution converges to a local minimum of the constrained non-linear problem [*Lecture notes in Optimization Theory. Chapter 4* 2020].

Optimization Speed The type of systems where a MPC could be applied is highly dependent on the time required for solving the optimization problem. Typically, slow solving times restricts the applications of MPC to systems with slow dynamics. Several methods and strategies for fast solving of QPs in a MPC context are presented in [Wang and Boyd, 2010]. The strategies presented include methods for exploiting the structure of the problem, making approximations as well as warm stating. By implementing all these methods, a solution to the MPC optimization problem can be achieved roughly 100 times faster than with a generic solver. Using these methods the typical computational time for one time-step in a MPC with 12-states, 3-inputs and a 30 step prediction horizon would be around 5 ms for 3 GHz AMD Athlon running Linux.

In MPC, warm start is the procedure of exploiting the result from the previous time-step optimization as the initial guess for the current optimization. This can be easily achieved by shifting the old solution one step along the prediction horizon, throwing away the current state and propagate the system dynamics on step at the further end of the horizon. This is similar to what is done by the SQP within one time step, with the difference of the warm-start using the result from the previous time step rather than the result from the previous QP iteration.

MPC for Autonomous Landing

For the purpose of landing the MAV on the boat, the MPC could be simultaneously used for control (sending commands to the to the low level controllers) and path-planning. In this section, the MPC formulation, cost matrices and constraints used for the landing algorithm, as well as generation of the reference trajectory will be presented.

As mentioned earlier, the nonlinear MPC-formulation has to be used in order to maintain the nonlinear dynamics of the simplified MAV model. By using as few non-linearities as possible in the optimization formulation, the SQP heuristic might prove a useful alternative for solving the optimization problem in Equation (4.33) efficiently. This requires linearization and discretization of the simplified MAV model, which can be done with standard methods such as a first order Taylor expansion for linearization and zero-order-hold for discretization. However, for the purpose of simulation Matlab's global solver *fmincon* was used. Thus the simplified MAV model was not linearized, only discretized. Recalling the system dynamics of the simplified MAV model

$$f(\mathbf{x},\mathbf{u}) = \begin{bmatrix} v_s cos(\gamma_s) cos(\chi_s) \\ v_s cos(\gamma_s) sin(\chi_s) \\ -v_s sin(\gamma_s) \\ a_s \\ \frac{1}{\tau_a}(a_s^{cmd} - a_s) \\ \frac{1}{\tau_\gamma}(\gamma_s^{cmd} - \gamma_s) \\ \dot{\chi}_s \\ -\tau_{\chi_1}\dot{\chi} + \tau_{\chi_0}(\chi_s^{cmd} - \chi_s) \end{bmatrix} \tag{4.34}$$

the discretization was performed with first order Euler discretization, using the MPC timestep h_{mpc} as follows

$$\mathbf{x}(k+1) = f_d(\mathbf{x}(k),\mathbf{u}(k)) = \mathbf{x}(k) + h_{mpc} \cdot f(\mathbf{x}(k),\mathbf{u}(k)) \tag{4.35}$$

where

$$\mathbf{x}(k) = \begin{bmatrix} x_s(k) & y_s(k) & z_s(k) & v_s(k) & a_s(k) & \gamma_s(k) & \chi_s(k) & \dot{\chi}_s(k) \end{bmatrix}^T \tag{4.36}$$

and

$$\mathbf{u}(k) = \begin{bmatrix} a_s^{cmd}(k) & \gamma_s^{cmd}(k) & \chi_s^{cmd}(k) \end{bmatrix}^T \tag{4.37}$$

Using this discrete model together with a quadratic objective function, linear inequality constraints, and setting $\bar{\mathbf{u}}_r(k)$ to zero, the nonlinear MPC formulation presented above reduces to

$$\min_{\mathbf{u}(k+i)} \quad \sum_{i=0}^{H_p-1} \|\mathbf{x}_r(k+i) - \mathbf{x}(k+i)\|_Q + \sum_{i=0}^{H_u-1} \|\mathbf{u}(k+i)\|_R$$

subject to

$$\begin{aligned} \mathbf{x}(k+i+1) &= f_d(\mathbf{x}(k+i), \mathbf{u}(k+i)), \quad i = 0 \ldots H_u - 1 \\ F_{x+i}\mathbf{x}(k) &\leq \mathbf{b}_x, \quad i = 0 \ldots H_p - 1 \\ F_{u+i}\mathbf{u}(k) &\leq \mathbf{b}_u, \quad i = 0 \ldots H_u - 1 \\ \mathbf{x}(k) &= \mathbf{x}_0, \end{aligned} \tag{4.38}$$

With well chosen penalties and inequality constraints as well as a reasonable reference trajectory this formulation will drive the MAV to align its trajectory along H_p with the corresponding boat reference-trajectory. In the end this incentive will lead to the MAV attempting to align itself with boat, both vertically and horizontally while also trying to match speed and course angle.

Reference Trajectory In order to use the MPC in this way, a reference trajectory for the boat must be available. Since the boat can not be controlled, and its trajectory is unknown and changing with time, some kind of approximation has to be made in each time-step. This is done by using an internal model of the boat dynamics, and then propagate the states forward in time given the current state. For each point along H_p the boat states are matched to the states in the simplified MAV model, inserting zeros for non-applicable states, forming $\bar{\mathbf{x}}_r(k)$. For the simulations of the landing, the simplified boat model presented in Section 2.4 is used for this trajectory generation. Forming the reference trajectory like this assumes knowledge of the current state ($\mathbf{p}_b$ and ψ_b) and inputs (δ_v and δ_ψ) to the simplified boat model. The position is provided by the positioning system used and the heading and control signals are assumed to be made available by sensors on the boat via the communication link. Here $\delta_v = v_b$ and $\delta_\psi = \dot{\psi}_b$ are assumed to be constant along the horizon. The assumptions made forming $\bar{\mathbf{x}}_r(k)$ should be reasonably precise if the boat is maneuvered with speed and angular velocity as constant as possible. Also, the reference trajectory corresponding to the earlier states on the horizon can be expected to be both the most important for the landing, and being better approximated than later states. This since the approximation of the boat states will

get worse when getting further into the future from the latest available measurement.

Cost and Constraints The final costs and constraints used were found by manual tuning in the simulation environment. The final Q and R penalty are

$$Q = \begin{pmatrix} 6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 6 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 9 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 35 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 5 \end{pmatrix} \tag{4.39}$$

$$R = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 20 \end{pmatrix} \tag{4.40}$$

and the limits for $\mathbf{x}$ and $\mathbf{u}$ used were defined by

$$F_{x+i} = \begin{pmatrix} I \\ -I \end{pmatrix} \in \mathbb{R}^{2n \times n} \quad \forall\, i \tag{4.41}$$

$$F_{u+i} = \begin{pmatrix} I \\ -I \end{pmatrix} \in \mathbb{R}^{2m \times m} \quad \forall\, i \tag{4.42}$$

and

$$\mathbf{b}_x = \begin{bmatrix} \mathbf{x}_{max}^T & -\mathbf{x}_{min}^T \end{bmatrix}^T \tag{4.43}$$

$$\mathbf{b}_u = \begin{bmatrix} \mathbf{u}_{max}^T & -\mathbf{u}_{min}^T \end{bmatrix}^T \tag{4.44}$$

where

$$\mathbf{x}_{max}^T(k) = \begin{bmatrix} inf & inf & 100 & 25 & 3 & 1 & inf & 0.25 \end{bmatrix} \tag{4.45}$$

$$\mathbf{x}_{min}^T(k) = \begin{bmatrix} -inf & -inf & -1 & 12 & -3 & -1 & -inf & -0.25 \end{bmatrix} \tag{4.46}$$

$$\mathbf{u}_{max}^T(k) = -\mathbf{u}_{min}^T(k) = \begin{bmatrix} 3 & 0.3 & inf. \end{bmatrix} \tag{4.47}$$

Notice that the -1 m lower bound for the vertical position does not mean that the MAV is allowed to fly under the water surface. This, as target was set 2 m over the water surface to reflect reasonable altitude where a human could catch the MAV.

Optimization The optimization was solved with Matlab's function *fmincon* using the *interior-point* algorithm with the parameter *MaxFunctionEvaluation*s set to 200. Also, gradients were specified for the objective function and the constraints.

4.5 Landing Strategy and Additional Decision Logic

A specific control algorithm might be well suited for the most parts of the landing sequence. Although, additional logic might be necessary in order to improve performance, increase safety and even make the landing feasible. This section will cover the strategies used for compensation of jitter and delays coming from sensors and communication as well as a method of suppressing undesirable MPC behaviour in the final part of the landing. A layer of logic that is not considered in this report is detection and decisions regarding when and how to a abort a landing. Even though this kind of logic would be highly desirable when attempting a real world landing, for the purpose of this thesis all landing attempts are pushed trough without considering abortion.

Sensor Jitter Compensation Strategy

If the successful sensor-readings are not periodic (as expected in LiDAR case) or sampled at a rate lower than the desired MPC update rate (as in the NEOM8N-GNSS case), it would be beneficial to use some strategy to minimize negative effects on the control performance. With the MPC aiming to run at some certain time interval, the strategy was to make use of the previously completed optimization results if no new sensor reading is available at the start time for the next optimization. If there are several consecutive periods passing without a sensor reading, the next control command at the control horizon can be used when a time interval has passed since the last control signal update. A flowchart of this scheme can be found in Figure 4.3. With this strategy the MPC optimization is only run if a relative position to boat is made available by the positioning system.

An alternative method could be to use dead reckoning to make an estimation of the boat position, and then make a new optimization every time step interval. Although, this approach requires the boat to continuously send sensor data or its estimated position found with the dead reckoning. Even though this approach was not used in the simulation, the assumptions mentioned could easily be fulfilled as long as there are relevant sensors on the boat as well as a working communication link between the MAV and the boat.

Delay Compensation Strategy

Naturally, there are delays in the system from sensor readings, communication and MPC computational time. These delays will hurt the performance of the landing algorithm since the distance between the MAV and the boat will not be the same

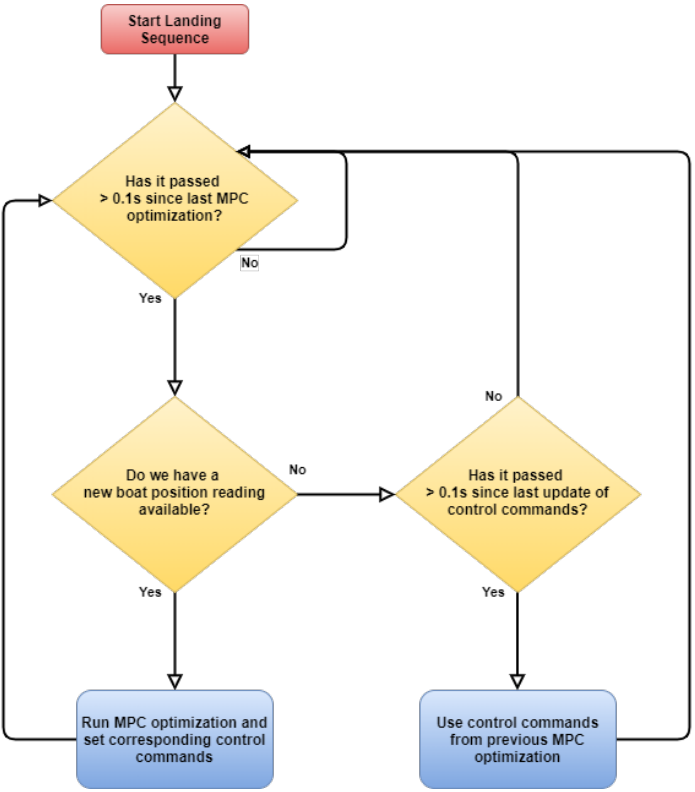


Figure 4.3: Illustration of the jitter compensation strategy. If no new data from the positioning system is available when it is time to start the MPC optimization, the inputs are taken from the control horizon of the previous solution.

when the sensor is read as when the MPC optimization is done. If the relative velocity is 10 m/s a total delay of 0.1 s would lead to a 1 m positional error in the direction of flight caused by the delay alone. In order to minimize the effects on the performance of the landing, a simple delay compensation was implemented. Using the simplified models for the MAV (see Section 2.2) and the boat (see Section 2.4), the states were extrapolated by propagating the system dynamics forward with correct time compensation for the MAV and the boat respectively. Note that the boat states should be compensated for the sensor, communication and MPC delays while the MAV states should be compensated for the MPC delay only. This strategy was implemented with the assumption of known and constant sensor delay and MPC computational time.

Vanishing Horizon

With the MAV moving faster than the boat, the MPC will react by trying to lower the velocity and align the MAV and boat trajectories as much as possible. If it is impossible for the MAV to slow down to the velocity of the boat (as will be the case most of the times) the MPC will do everything it can in order to slow down the MAV. It turns out that the MPC is doing this by letting the MAV trajectory oscillate heavily, making the flight distance longer. This behaviour is highly undesirable and needs to be avoided.

To counteract this behaviour, a simple algorithm refereed to as *vanishing horizon* is proposed. This is done by removing the MPC x and y-penalties from all states along the prediction horizon which are spatially in front of their corresponding state on the boat trajectory. With this strategy all incentive for the MAV to start oscillating goes away while keeping the desire to find the boat, stay at the correct altitude and minimize the relative MAV-boat velocity. An illustration of the principle can be seen in Figure 4.4.

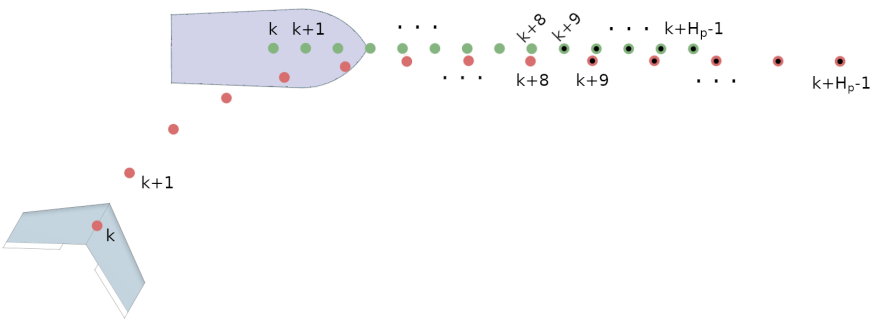


Figure 4.4: Illustration of the vanishing-horizon principle. Here, the red and green dots represent the prediction horizons for the MAV and the boat respectively. The x and y penalty weights of the MPC are set to zero for points $k+9$ to $k+H_p-1$. In the illustration, the final part of the MAV trajectory is shifted slightly downwards to make it easier to see the two trajectories.

Wind Compensation

The MPC states and constraints are defined in $\mathcal{I}$ (with $\mathbf{k}_z$ pointing up), while forces and moments acting upon the MAV, velocity limits, etc., are all dependent on the air-relative movement of the MAV. With wind magnitudes large enough to significantly separate $\mathbf{V}_g$ and $\mathbf{V}_a$ (see Sections 2.1 and 2.3), this is something that preferably should be taken into account when designing the control system. In order to make any compensations for wind, an estimation of the wind must be available.

This is the case when using the *Extended Kalman Filter* (EKF) in PX4 or Ardupilot, if the MAV is equipped with an airspeed sensor and a GNSS receiver.

To some extent, the effects of the winds are inherently counteracted by the integral action in the TECS and course controller respectively. Even though a change in direction might result in diverse responses under different wind conditions, the integral action makes is possible for the MAV to set path and course angles independently of the wind. Instead, the wind effect that most urgently must be addressed in the control system is the velocity inequality constraint in the MPC. The velocity-state of the MPC, and thus also the corresponding velocity inequality constraints, are, as mentioned above, defined in the inertial frame. The true limits of the MAV ground velocity should therefore, if possible, be adjusted for the wind in the direction of flight.

With this in mind a simple strategy was deployed to change the velocity inequality constraints given the current course and wind conditions. Every time, before the MPC is updated, the projection of the horizontal component of the wind vector ($\mathbf{V}_{w_a}$) onto the horizontal component of the MAV ground-speed vector ($\mathbf{V}_g$) is computed. The resulting value is the horizontal wind component in the direction of flight. This value is then added to the MPC velocity inequality constraint along H_p.

Wave Compensation

A simple strategy counteracting the effects of waves was also used. If the boat was to vertically oscillate, this could either be ignored or be incorporated in the landing strategy. Incorporation would require the MAV to time the periodicity of the boat and plan the intersection accordingly. However, in order to keep things simple, the wave periodicity was ignored and a simple 1^{st}-order low-pass filter was used for filtering of the vertical component of the estimated boat position. The filter used was the following

$$z_{b,filter}(k) = 0.98 z_{b,filter}(k-1) + 0.02 z_b(k) \tag{4.48}$$

Final Push

Since the MAV, in most cases, can not move as slowly as the boat, the landing is done by letting the MAV fly over or into the boat and being captured by a human being (see Sections 1.2 and 5.2). Although, in a few scenarios with a lot of headwind present, it might be possible for the MAV to move at the same or even lower velocity than the boat, leading the MAV to slowly aligning itself with the boat. This might lead to the MAV not passing the boat and thus not clearly define a point where the landing is completed and the offset from target can be recorded. In order to avoid this situation, a minimum threshold was set on velocity used for calculation of the boat reference trajectory, pushing the MAV forward a little extra if needed.

Parameter	Identified Value	Used Value
τ_a	0.111	0.056
τ_γ	0.601	0.076
τ_{χ_0}	2.371	10.3619
τ_{χ_1}	6.663	133.26

Table 4.1: Identified and used parameters for the simplified MAV model. Notice that there is significant difference in the corresponding values.

4.6 System Identification

In order to maximize the MPC performance, the model used in the MPC should reflect the real behaviour of the plant as much as possible. To find initial parameter values of the model, a simple system identification was made. The parameters that need to be identified from the simplified MAV model (see Section 2.2) are τ_a, τ_γ, τ_{χ_0} and τ_{χ_1}, The corresponding systems are two 1^{st} order input-output system for the acceleration and flight path angle, and a 2^{nd} order input-output system for the heading angle, all with unit gain.

The identification was made in the simulation using using the closed loop system of the MAV and the low-level controllers (pitch, roll, course and TECS). At a nominal airspeed of 15 m/s, an isolated step input was sent to each of the inputs of the TECS (a and γ) and to the course controller (χ) respectively. For each attempt the corresponding state of interest was recorded and saved. The input-output data was then fed to Matlabs *tfest* function for identification of the sought parameters.

For acceleration and flight path angle the step-responses are expected to behave differently for positive and negative input steps. This as the mechanisms driving positive and negative quantities are not symmetrical. For example, in the acceleration case the sign will be decided by the balance between the propulsion force and the total drag force, two forces unlikely to be symmetrical. Thus, the data recorded for identification of τ_a and τ_γ was generated by negative step responses, to better capture and approximate the dynamics of a landing sequence.

The step responses for the MAV and corresponding response for the identified parameters can be seen in Figures 4.5 - 4.7 below. The identified parameters can be seen in Table 4.1. From the plots in Figures 4.5 - 4.7 it can be seen that the step responses are far from perfect. I might be argued that the γ input-output behavior should be modeled with a 2^{nd} order transfer function, similarly as was done for χ. Another factor that might have significant effects on the correspondence between the identified parameters and the behaviour in the simulation is whether or not the control surfaces are getting saturated.

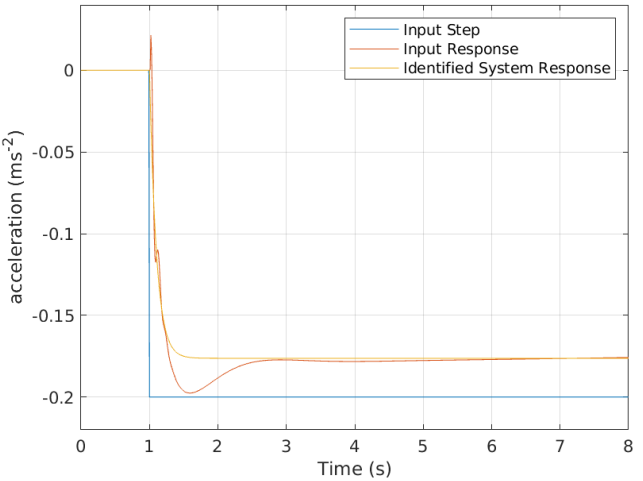


Figure 4.5: Step response for acceleration (a) and the corresponding identified system.

As one might expect from the step responses, the found parameters did not work very well in practice. Instead the parameters were manually tuned with the identified values as a starting point. In the end, the parameters used significantly differed from the identified ones. Another thing worth noticing is that input responses vary significantly with airspeed (V_a). This behaviour comes as no surprise as the aerodynamic forces and moments modeled are dependent on airspeed.

Parameter Interpretation The parameters take part in describing the MPC's belief on the MAV response to certain control control commands. For example, a small value of τ_a or τ_γ implies a faster believed change in system state for a given control command. With this in mind the MPC will output smaller control commands in order to predict the same trajectory. This might lead to a less aggressive controller. However, one cannot forget that this is also coupled with the penalty of the control signal. If the penalty on the control signal stays the same, a belief of a higher gain makes it cheaper to lay out a more aggressive trajectory.

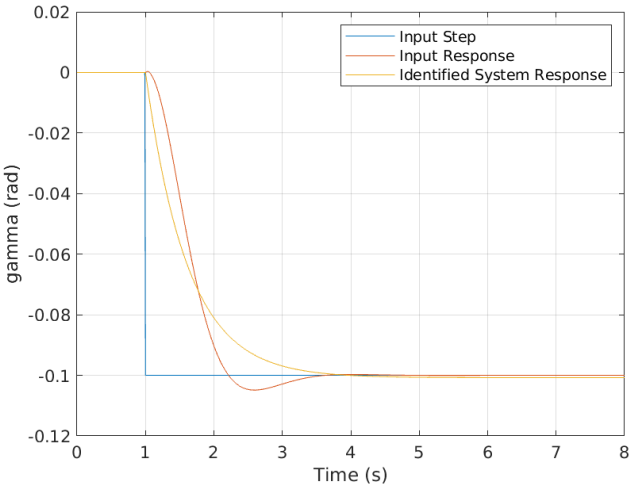


Figure 4.6: Step response for flight path angle (γ) and the corresponding identified system.

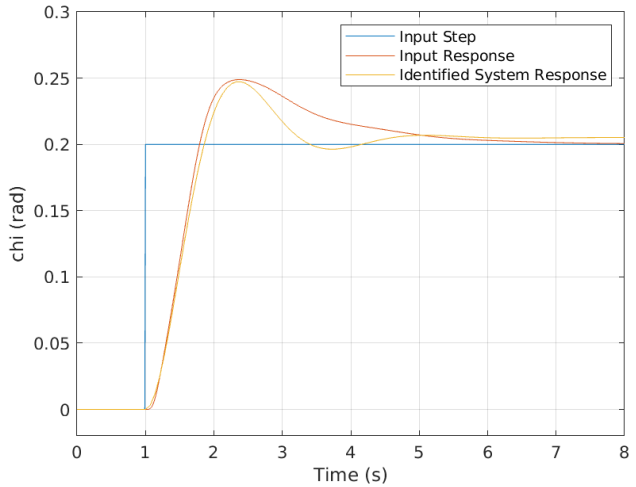


Figure 4.7: Step response for course angle (χ) and the corresponding identified system.

5

Simulation

The simulation combines the system models of the MAV and its control systems, the boat, positioning systems, wind, and waves. The purpose is to replicate the real target scenario in order to analyze the performance of the positioning systems in combination with the control algorithms and the impact of external factors. The simulation was implemented in Matlab.

Each simulation represents a landing attempt, where the MAV is initialized at some position behind boat. The MAV then navigates towards the boat with the control strategy for autonomous landing, using information from the simulated sensors for relative positioning. The simulations were run for the different sensors systems, each one from a number of combinations of initial positions and environmental conditions. In all scenarios the heading of the boat was constant. Additionally the initial state and control signals of the MAV were trimmed such that the initial course remains constant with no need of additional control commands, also taking ambient wind into account.

5.1 Discretization

Most of models used are described by continuous time differential equations and had to be discretized in order to work with the discrete time simulation. To keep this step as simple as possible, a first order Euler discretization was used, i.e., for a continuous time system

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}) \tag{5.1}$$

the corresponding discrete time system is

$$\mathbf{x}_{t+1} = \mathbf{x}_t + h \cdot f(\mathbf{x}_t, \mathbf{u}_t) \tag{5.2}$$

where h is the time between two consecutive discrete time steps. For the simulations, h was set to 0.01 s.

5.2 Assumptions

The number of possible MAV-boat relative starting positions in combination with all possible weather conditions is large. In order to reduce the number of simulations and focus on the most likely and realistic scenarios, assumptions on each subsystem in the simulation were made. These assumptions and corresponding underlying reasoning are described below.

Wind and Waves

In the archipelago where most accidents occur, the dominant factor in waves forming is wind. This, since the sways present at open sea is broken up by small islands and cobs. Dealing with mostly wind-generated waves, it was assumed that wind and waves always are moving in the same direction and that their magnitudes are somewhat correlated. For zero wind, zero waves was assumed. For wind magnitudes of around 10 m/s, two wave amplitudes, 0.5 m and 1 m was used. For all combinations of wind and waves used in the simulation, see Tables 5.3 and 5.4 in Section 5.3.

MAV

At the start of the simulation the MAV is commanded to fly in a straight line until the first successful position sensor reading. The line is defined by the MAV's initial course angle, altitude and air-relative speed. The control actuators, angle of attack and side-slip angle are trimmed in order for the MAV to continue flying along the initially defined path without changes in the control commands. The trimming also takes ambient wind into account and makes the corresponding compensations needed.

Boat

In general, the landing is easier if the relative velocity between the MAV and the boat is as small as possible (as long as the MAV can fly fast enough to reach the boat within reasonable time). For example, in [Persson, 2019] the landing strategy is based on aligning the MAV and the target in the horizontal plane while making the approach from above. To some extent, this strategy makes the assumption of the MAV and boat being able to move equally fast. A small relative velocity in the horizontal plane provides a larger time window where the approach from a above can be made. In our case this might not be easily achievable since the nominal speed of the MAV is significantly higher than regular boat speeds. Even though there is an overlapping speed-interval for low MAV speeds and high boat speeds, it is also convenient if the boat is moving as slowly as possible. This, since a slower boat speed makes less obstacles and other interfering factors come into play.

As mentioned before, the velocity limitations of the MAV relates to the surrounding air rather than to the ground. In order to minimize the relative velocity without the

MAV moving slower or the boat faster, the boat can turn up against the wind and make the landing take place in headwind. Thus, an assumption on the boat moving straight or almost straight against the wind was made.

Waves are another external factor which might limit the velocity of the boat. Ideally the some kind of relationship between the weather conditions and a reasonable boat velocity should be established. However, this was not done and the boat velocity was set to 5 m/s for all simulation cases.

Positioning Systems

The GNSS and 3D LiDAR positioning systems were simulated with their specified update rate, delay and positioning error/dynamics. The sensors are assumed to be sampling at a constant rate without jitter. Although, for example in the LiDAR case, updates can still arrive with different time intervals since it possible for the sensor to sample without getting a reading of the MAV position (if there is no laser beam hitting the MAV). It was also assumed that the delay from the actual time of sampling until the reading being available to the MAV, is both constant and known to the MAV. Finally the positioning error was simulated as described in Sections 3.5 and 3.6.

MPC

Based on the discussion of computational time for solving QPs and assuming the MPC optimization is solved with SQP, the MPC was simulated to have 30 ms execution time in the simulation. The time-step used in the MPC (h_{mpc}) was set to 100 ms.

5.3 Simulation Cases

Positioning Systems

The sensors used for the simulations was mainly of two types, GNSS and LiDAR. The GNSS sensors used was the NEO-M8N and ZED-F9P described in Section 3.5 as well as a 'perfect' GNSS sensor with zero delay and zero positioning error, used as a reference. The 3D LiDAR sensors simulated are Velodyne Puck, Velodyne Alpha Prime and Ouster OS1 as described in Section 3.6, representing LiDAR units in different segments of price and performance. A summary of the simulated sensors can be seen in Table 5.1.

Initial Conditions

For the simulations a few initial states of the MAV in relation to the boat were chosen. Ideally the the MAV starts the landing sequence right behind the boat with its

Sensor			
Sensor	Type	Update rate (Hz)	Delay (ms)
PERFECT	GNSS	100	0
NEO-M8N	GNSS	5	25
ZED-F9P	GNSS	20	25
Velodyne Puck	LiDAR	5	50
Velodyne Alpha Prime	LiDAR	5	50
Ouster OS1	LiDAR	10	50

Table 5.1: Update rate and delay for each sensor used in the simulation.

course directly towards the boat. However, this condition is not anything the operator would like to spend a lot of time to achieve. In light of this, a few different MAV-boat relative starting conditions were tested, with the common factor of the MAV being heading approximately towards the boat. The different starting conditions of the MAV can be seen in Table 5.2.

MAV Starting Conditions in relation to the boat					
Pos. #	$x_m^{\mathcal{V}_{1boat}}$ (m)	$y_m^{\mathcal{V}_{1boat}}$ (m)	$z_m^{\mathcal{V}_{1boat}}$ (m)	$\chi^{\mathcal{V}_{1boat}}$ (rad)	V_a (m/s)
Pos. 1	-80	0	-5	0	17
Pos. 2	-80	8	-5	0	17
Pos. 3	-80	8	-5	-π/12	17
Pos. 4	-80	-8	-5	0	17
Pos. 5	-80	-8	-5	π/12	17

Table 5.2: Starting conditions for the MAV used in the simulation. Notice that the states x_m, y_m, z_m and χ are expressed in the boat veichle-1 ($\mathcal{V}_1$) frame.

Wind and Waves

For each combination of positioning sensor and MAV starting position, three wind-wave scenarios were simulated. The different wind-wave combinations used can be seen in Tables 5.3 and 5.4.

Wind Parameters									
Scen. #	$w_{x_a}^{\mathcal{V}_{1boat}}$	$w_{y_a}^{\mathcal{V}_{1boat}}$	$w_{z_a}^{\mathcal{V}_{1boat}}$	L_u	L_v	L_w	σ_u	σ_v	σ_w
Scen. 1	0	0	0	-	-	-	0	0	0
Scen. 2	-10	0	0	200	200	50	4.06	4.06	4.06
Scen. 3	-7	-7	0	200	200	50	4.06	4.06	4.06

Table 5.3: Wind parameters for the three different environmental scenarios. Notice that w_{x_a}, w_{y_a} and w_{z_a} are expressed in the initial boat veichle-1 ($\mathcal{V}_1$) frame.

Wave Parameters						
Scenario #	A	$\psi_{wave}^{\mathcal{V}_{1boat}}$	λ	T	ϕ_{wave}	σ_{P_w}
Scenario 1	0	0	10	2.5	0	0
Scenario 2	0.5	π	10	2.5	0	0
Scenario 3	1	$-\pi/4$	10	2.5	0	0

Table 5.4: Wave parameters for the three different environmental scenarios. Notice that ψ_{wave} is expressed in the inital boat veichle-1 ($\mathcal{V}_1$) frame.

6

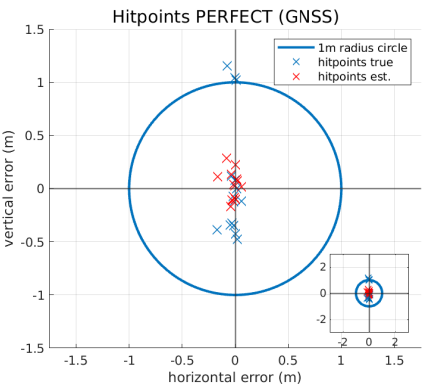
Results

The results of the simulation are evaluated by the ability of the MAV to reach a certain landing point tied to the boat. This ability is measured by the MAV lateral and vertical offset from the target point when it passes the boat. The landing target point reflects the position at which a crew member would stand and catch the MAV. Other factors relevant to the landing quality might be the total time of the landing or the relative velocity at the time of landing. However, those metrics were not included in the evaluation of the landing in order to limit the results to the most important factors of a successful landing.

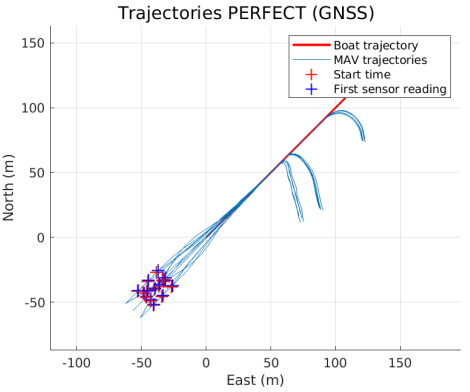
The results consist of two plots for every tested positioning system. Firstly, the target grouping plot shows how well the MAV was able to reach the desired position. The red crosses in these plots show the estimated hitpoints, showing the controller belief of the relative position between the MAV and the boat when the controller thinks the MAV crosses the boat. The blue crosses show the true relative position when the MAV actually crosses the boat. Secondly, the results also contain trajectory plots, showing the trajectories of the MAV during the landings. The starting position of the MAV in each simulation case is marked in red, and the position where the first sensor reading occurs is marked in blue. When a landing is completed by the MAV having passed the boat, commands are given to gain altitude and turn right. This, resulting in a hook at the end of the trajectory for a finished landing attempt.

6.1 Evaluation of Landing Algorithm

In order to separate behaviours dependent on the landing strategy itself and the sensor used, the first results presented are from landings where a PERFECT sensor was used. Here, the error and delay is set to zero and the refresh rate is the same as the simulation time-step. The target grouping plot and trajectory plot can be seen in Figure 6.1.



(a) Target grouping plot.



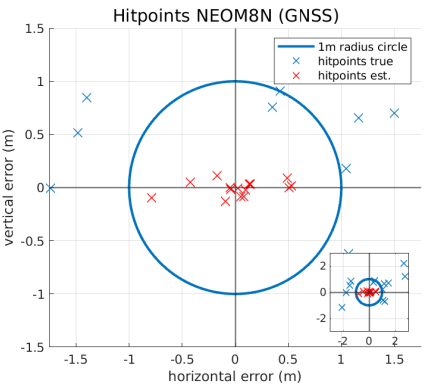
(b) Landing trajectories.

Figure 6.1: Target grouping plot and landing trajectories for the MAV when using the PERFECT position sensor.

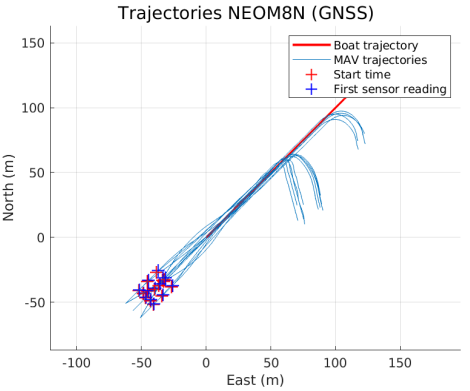
6.2 GNSS

NEO-M8N

The target grouping plot and trajectory plot for the simulation case when using a NEO-M8N GNSS device can be seen in Figure 6.2.



(a) Target grouping plot.

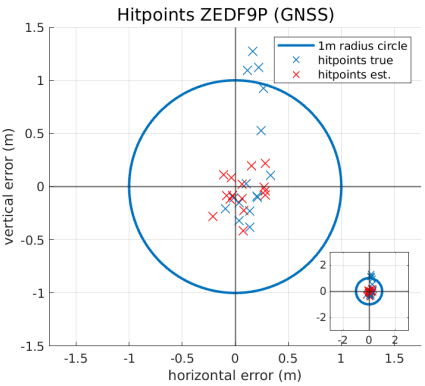


(b) Landing trajectories.

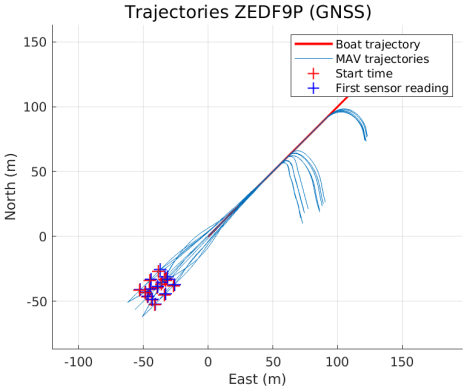
Figure 6.2: Target grouping plot and landing trajectories for the MAV when using a NEO-M8N GNSS device.

ZED-F9P

The target grouping plot and trajectory plot for the simulation case when using a ZED-F9P GNSS device can be seen in Figure 6.3.



(a) Target grouping plot.



(b) Landing trajectories.

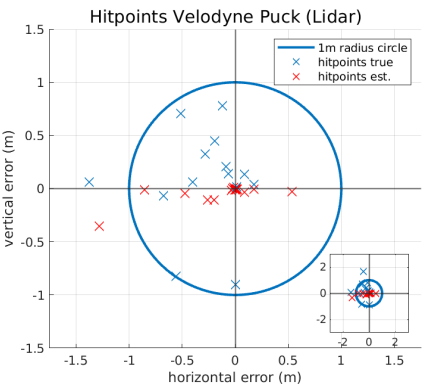
Figure 6.3: Target grouping plot and landing trajectories for the MAV when using a ZED-F9P GNSS device.

6.3 3D LiDAR

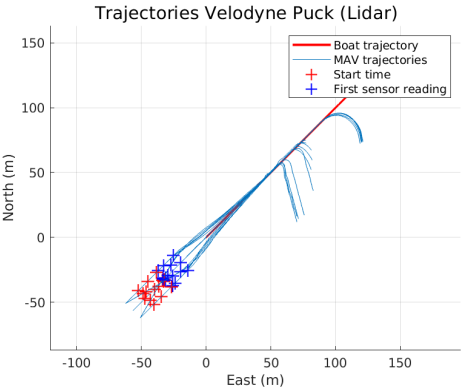
When running the simulations with a 3D-LiDAR, scenarios where the MAV never enters the volume in which it can be detected by the LiDAR might arise. This will result in the MAV flying in its initial direction without trying to reach the boat, in turn leading to straight lines in the trajectory plots (no hook at the end).

Velodyne Puck

The target grouping plot and trajectory plot for the simulation case when using a Velodyne Puck 3D LiDAR device can be seen in Figure 6.4. The detection range limit for the 3D LiDAR was set to 60 m.



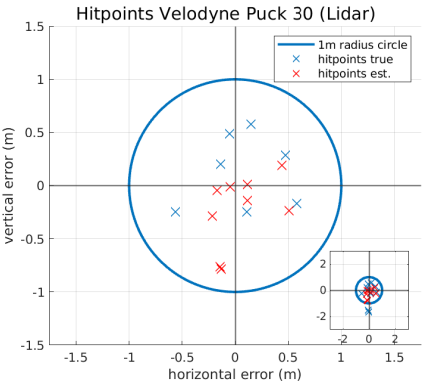
(a) Target grouping plot.



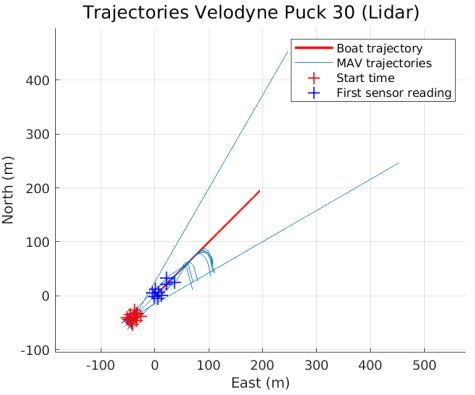
(b) Landing trajectories.

Figure 6.4: Target grouping plot and landing trajectories for the MAV when using a Velodyne Puck 3D LiDAR device with 60 m detection range limit.

The results for the same simulation cases as above, but with a 30 m detection range limit, can be seen in Figure 6.5.



(a) Target grouping plot.

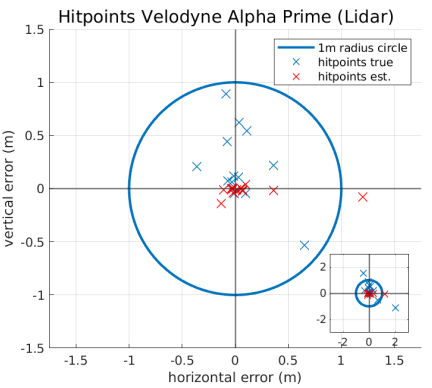


(b) Landing trajectories.

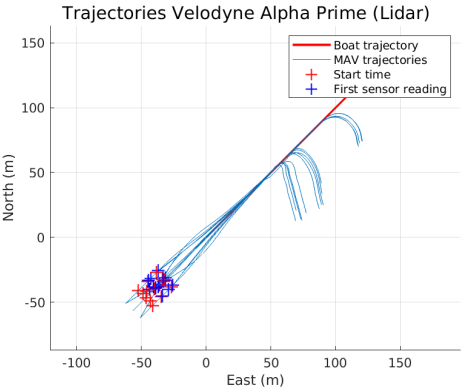
Figure 6.5: Target grouping plot and landing trajectories for the MAV when using a Velodyne Puck 3D LiDAR device with 30 m detection range limit.

Velodyne Alpha Prime

The target grouping plot and trajectory plot for the simulation case when using a Velodyne Alpha Prime 3D LiDAR device can be seen in Figure 6.6. The detection range limit for the 3D LiDAR was set to 60 m.



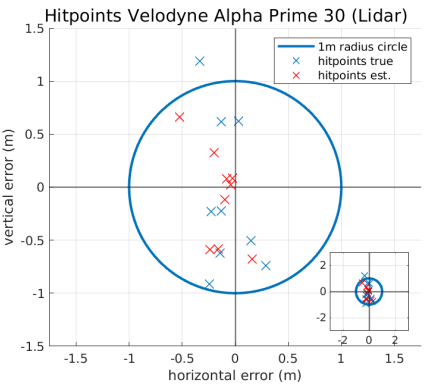
(a) Target grouping plot.



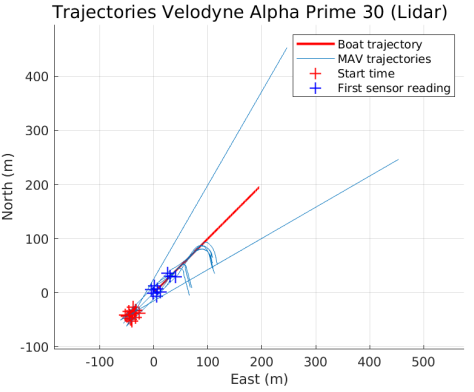
(b) Landing trajectories.

Figure 6.6: Target grouping plot and landing trajectories for the MAV when using a Velodyne Alpha Prime 3D LiDAR device with 60 m detection range limit.

The results for the same simulation cases as above, but with a 30 m detection range limit, can be seen in Figure 6.7.



(a) Target grouping plot.

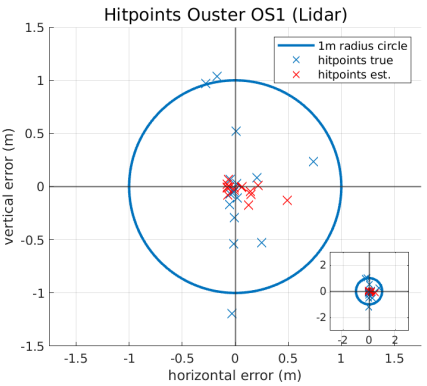


(b) Landing trajectories.

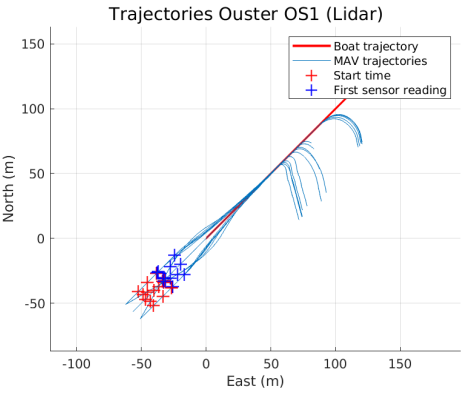
Figure 6.7: Target grouping plot and landing trajectories for the MAV when using a Velodyne Alpha Prime 3D LiDAR device with 30 m detection range limit.

Ouster OS1

The target grouping plot and trajectory plot for the simulation case when using an Ouster OS1 3D LiDAR device can be seen in Figure 6.8. The detection range limit for the 3D LiDAR was set to 60 m.



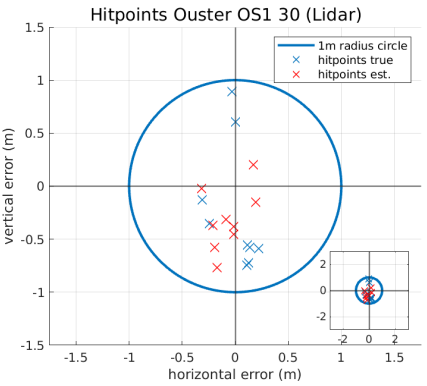
(a) Target grouping plot.



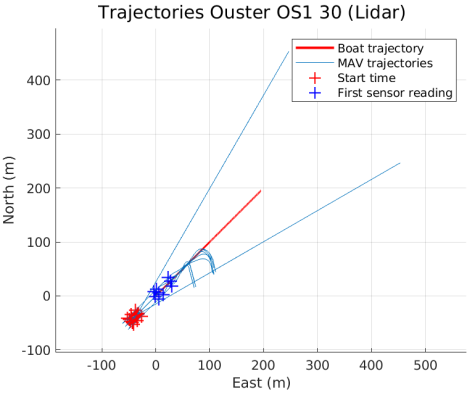
(b) Landing trajectories.

Figure 6.8: Target grouping plot and landing trajectories for the MAV when using an Ouster OS1 3D LiDAR device with 60 m detection range limit.

The results for the same simulation cases as above, but with a 30 m detection range limit, can be seen in Figure 6.9.



(a) Target grouping plot.



(b) Landing trajectories.

Figure 6.9: Target grouping plot and landing trajectories for the MAV when using an Ouster OS1 3D LiDAR device with 30 m detection range limit.

7

Implementation

7.1 PX4 & ROS

The original problem formulation in Section 1.3 included implementation on a real-world system and thus the control system naturally had to be transferred to the indented platform. But as previously mentioned, due to time constraints, this was never done. However, before coming to this insight, some parts were implemented, for example the TECS and course controllers described in Section 4.2.

As discussed in Section 4.1 the target platform is the PX4 flight stack controlled by a companion computer running ROS. Using the PX4 *softwar§e in the loop* (SITL) simulation in Gazebo (a robot simulation tool) together with MAVROS and the PX4 offboard mode, those controllers could successfully interact with the flight stack as intended. The performance of those controllers was never evaluated but the behaviour seemed reasonable and the successful interaction with PX4 trough ROS was a valuable proof-of-concept.

7.2 Small Scale Test Vehicle

Because a GNSS positioning system was available and ready to use, a real life test with a vehicle navigating with input from relative positioning between two receivers, a small scale test vehicle, seen in Figure 7.1, was prepared. A test case was set up where the car was intended to drive from an initial position of around 30 m away from, and heading away from, another stationary GNSS receiver streaming its position to the vehicle according to the suggested implementation presented in Section 3.5.

The vehicle used was a 1:10 scale RC car, equipped with a Vedder Electronic Speed Controller (VESC) that enables it to drive steadily forward at slow velocities. A Raspberry Pi computer on board the car was used for reading sensors, estimating states, processing data and producing control output. A compass module was added

to enable course measurement.

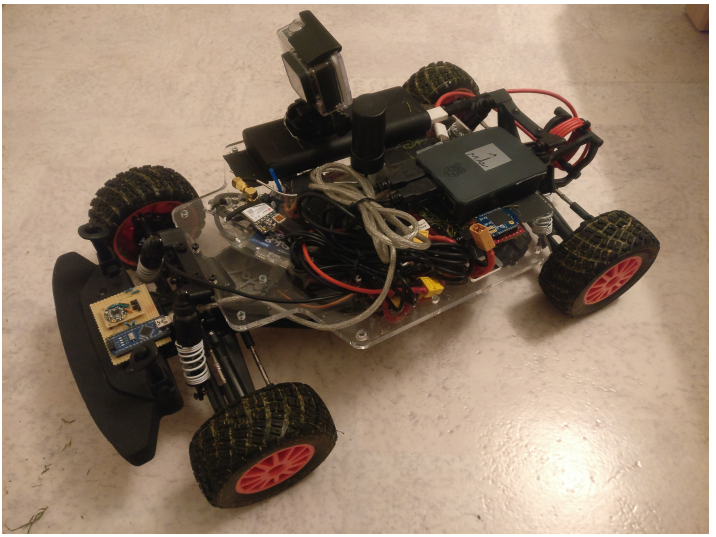


Figure 7.1: Small scale test vehicle.

Due to time constraints, the control system used was not MPC-based as in Section 4.4, but a simpler P-controller for course control was used. As the main purpose of the test was to validate whether the isolated GNSS tests made in Section 3.5 would apply to moving receivers, this simple control system was considered sufficient.

The test sequence, where the vehicle drove with a velocity of around 2 m/s, was run 7 times before the system encountered battery issues. The results of the test showed that the car was able to navigate towards the remote GNSS station and always hit the target within approximately 0.3 m. A freeze frame of every test run shortly before the vehicle hit the remote station (covered by a blue backpack) can be seen in Figures 7.2 through 7.5.

(a) Test run 1.

(b) Test run 2.

Figure 7.2: Endpoint for the test vehicle at run 1 & 2.

(a) Test run 3.

(b) Test run 4.

Figure 7.3: Endpoint for the test vehicle at run 3 & 4.

(a) Test run 5.

(b) Test run 6.

Figure 7.4: Endpoint for the test vehicle at run 5 & 6.

(a) Test run 7.

Figure 7.5: Endpoint for the test vehicle at run 7.

8

Discussion

In this chapter, discussions related to the previous chapters of the thesis are made. This is then continued with a section on conclusions drawn from the project, followed by a final section on future work including suggestions on steps to take the project further, in the future.

Throughout the project many assumptions were made in order to simplify a complex problem. Naturally all of the assumption make the results of the simulation less reliable. The natural way of finding out how well the landing strategy combined with the different positioning systems actually is working, would be to start making more real life tests. Even though the initial intention was to do exactly so, the many aspects and complexity of the problem together with time constraints put a stop to those plans.

8.1 Modeling

In general, most aspects of the modeling could have been done with more detailed models. A natural step for increasing the complexity of the modeling is to move to a flight simulator. Nevertheless, as all models used were implemented directly in code, full control of the mathematical models used was obtained.

Starting with the MAV, the modeling was done with an aerodynamic model capturing most of the behaviour characteristics of a flying wing (Zagi) around its normal point of operation. The parameter-set used for the MAV is based on a different flying wing than the one that will be used in real life situations, introducing an element of uncertainty regarding the dynamic and kinematic behavior of the MAV during flight. Since there is no aerodynamic parameter set available for the newly designed airframe, using the parameters for the Zagi airframe could probably be considered one of the better approximations available for desired MAV behaviour.

The parameters used in the Dryden wind model are taken directly from literature. Therefore, it is hard to tell how realistic these are for the particular cases of simulation. For simulation purposes it would have been interesting to identify the magnitude of these parameters for a few different typical weather conditions and locations along the coastline. One, possibly major, factor regarding winds was not modeled; the turbulence in the air created by the boat. In real life this factor will comes into play close to the boat, where the most critical part of the landing takes place, and might have large effect on forces affecting the MAV. For sure this is something that has to be evaluated carefully, either by literature and simulation or real life testing. This turbulence could be a factor that makes the proposed landing impossible. A few intuitive alternative approaches would be to either minimize the velocity of the boat in relation to the surrounding air, or try to approach the boat from the top or the side. Although, such approaches would require more from the landing strategy as the time window for a possible landing would decrease significantly if the MAV can not fly as slowly as the boat moves.

In general the wave model is simplified. In order to make the model more realistic, the sinusoidal wave could have been replaced by a trochoidal wave, which more accurately resembles ocean waves. Another improvement would be to add a more realistic stochastic behaviour on top of the deterministic part of the wave.

Closely related to the modeling of waves is the modeling of the hydrostatic and hydrodynamic interaction between the boat and the water. The model of the boat is simplified and only contains two-dimensional kinematics, except for the vertical position component driven by the wave behavior. Roll and pitch behavior was not included into the model because of down prioritization due to time constraints. This simplification imposes several constraints onto the conclusions that can be drawn on the full system simulation. Most notably, the changes in the 3D LiDAR vertical FoV due to variations in the boat orientation is not taken into account in the case where the 3D LiDAR is mounted fixedly onto the boat. This should be considered an important factor as it with time changes the limits of where the MAV can be detected. This can be expected to limit the utility of a fixedly mounted 3D LiDAR positioning system in wavy conditions.

8.2 Positioning

The analysis of the influence of weight addition on lost flight time on the MAV is based on the parameter set of the Zagi airframe and is therefore not directly applicable to the airframe that will be used in the end. Differences in what velocities the airframes are optimized for make the parameters for lift and drag forces vary, which in turn makes an influence on how additional weigh affects the flight-time.

There is no guarantee that the design criteria has been similar for the two airframes or that the calculated value should be used as a reference figure. The calculated value is also valid only for small changes in added weight, as the drag and lift coefficients tend to be nonlinear over larger intervals of angle of attack. Even so, the 9 seconds of lost loitering time per added gram of weight suggests that there is room for small scale devices with low weight without sacrificing any significant flying time.

Communication

The results of the radio communication link tests were promising both in isolated range and delay tests, as well as in the implementation with the small scale test vehicle. There was both a long enough communication range to connect at a large enough distance, as well as a small enough delay for the full system simulation to avoid instabilities. However, there are more ways that the information can be communicated from boat to MAV which have not been explored.

GNSS

In the simulations, the GNSS errors recorded from testing were used as the position error. The conditions that the tests were performed in is not directly applicable to the conditions that the MAV is intended to fly in eventually, even though the tests were performed around water. Since the devices remained still throughout the tests, and no analysis has been made into any delays within the process of measurement in the GNSS device there might be an additional delay that has not been taken into account.

The information returned by the GNSS modules is the difference in latitude, longitude and altitude by the two GNSS modules. Together with information about the heading of both vehicles, it is well suited for a controller such as an MPC. The tests performed on the GNSS devices are mostly related to performance, where the test results as well as small scale test implementation indicate that the precision, refresh rate and delay are within, or close to, the required limits when using the ZED-F9P device pair. Since the devices use radio frequency, there is little influence of rain, fog or other environmental factors which make them robust in many weather conditions. Since the configuration of the GNSS setup requires data to be communicated, there is need for a communication link which adds to the complexity of this solution.

3D LiDAR

The 3D LiDAR method was difficult to evaluate as no physical device was available for testing purposes. It is, however, an interesting method to evaluate because of the surge in popularity in recent years and the increase in availability of higher performing devices. In hindsight it is obvious that the analysis made on the method contains a large amount of assumptions, simplifications and a model that requires

quite a large amount of validation to be considered useful.

A number of simplifications have been made in the simulation of the 3D LiDARs, some of which are mentioned in Section 3.6. It is assumed that no other objects than the MAV are visible to the 3D LiDAR. In reality there might be other objects like bird, boats and similar things that come into the field of view and cause position ambiguities. Looking deeper into object detection strategies would be beneficial for this purpose.

The difficulty of modeling the 3D LiDAR behavior, without access to an actual device for testing, lies in approximating how far away the MAV can be detected. As presented in the the LiDAR section, rain, object material, surface, color and orientation affect the maximum distance and determining this value is difficult.

The information returned by the 3D LiDAR is the position of the MAV expressed in the body frame of the boat. Using information about the orientation as well as the course of the boat and the MAV, the relative position can be used in the same way as a GNSS measurement. As LiDAR is based on light, there is a risk using the system in rainy conditions. The accuracy of LiDAR is relatively high, with decimeter or in some cases centimeter precision. The 3D LiDAR relative positioning method would require data to be communicated to the MAV in real time which makes it necessary to facilitate the communication. The method cannot be considered a simple system as it comprises many sub modules, considerable processing of data and potentially implementation of complex object detection strategies.

ILS

Two fundamental flaws exist with the explored implementation of the small scale ILS system using two directional antennas and RSSI. Firstly, the antennas have to be mounted in a stabilized way in order to avoid two consequential issues, vibrations that cause additional noise to the measurements at long distances, and changes in antenna direction due to variations in the boat orientation. Secondly, directional antennas have side lobes that would create heavily irregular measurement patterns when approaching close to the boat. The real ILS system can avoid these problems by having the antennas positioned at the far end of the runway. The side lobe issue can potentially be attenuated by considering different type of directional antenna.

8.3 Autonomous Landing

In general, the landing algorithm is working pretty well in the simulations. There is likely room for improvement, specifically in terms of getting the controller to be more aggressive under some circumstances while being less so under others. This problem is believed to arise from the model mismatch between the full and the

simplified MAV model.

Starting from the bottom with the low level controllers for pitch and roll, they seem to be doing their jobs. Those controllers would not be used during a implementation in PX4 or Ardupilot as the corresponding functionality is already implemented an ready to use in those flight stacks. By looking at the step response in Figure 4.7 in Section 4.6, the course controller itself seems to behave reasonably well, and something similar could probably be used successfully. Similarly, based on Figures 4.6 and 4.5, the TECS is also doing what is should reasonably well, with some reservation for the somewhat strange behavior of the step response of the acceleration. Based on the work in [Argyle and Beard, 2016], the TECS could probably be improved to some extent by using a more advanced version of the controller.

Moving on to the MPC, this is a place where a few major improvements possibly could be made. In general the MPC seems to be working quite well, but it took bit of tuning to get there. The major problem was finding a balance between turning aggressively enough when not facing directly towards the boat, while avoiding oscillations when doing so. A reason for this could be differences between the full MAV model and the model used by the MPC. A model matching the real plant as closely as possible is essential for the performance of the MPC, and it is clear from the parameter identification in Section 4.6 that the MPC internal model is a little bit off from the full MAV model. Solving a problem in a optimal way may sound very good, but does not necessarily have to be so. Specifically if the problem solved by the optimization is not the true problem, which partially seems to be the case for the model used in the MPC. A simple idea that could be used to extend the MAV-model used by the MPC is to expand the path angle behaviour to a 2^{nd} order system, similarly as for the course angle. Another idea of interest could be to somehow incorporate the airspeed into the model, as the airspeed has quite large effects on the MAV response to control commands. Another area of improvement could be to address the wind in a better way. Successfully incorporating magnitude and direction of the wind directly into the model could potentially help the MPC to utilize wind to its advantage.

However, a more detailed model of the plant used by the MPC, will in most cases result in additional states as well. More states will lead to longer computational time or more computational power needed. This is trade-off that has to be considered if the MPC-model is to be expanded further. The speed of solving the MPC optimization on the indented CPU should be analysed in order to get an idea how much the MPC model could be expanded before causing a computational bottleneck. From the discussion in [Wang and Boyd, 2010] it seems like adding a few more states should be possible without causing too much trouble. The optimization step itself could also have been done differently, for example by adding slack variables allowing for soft constraints, or formulating the optimization in terms of $\Delta\mathbf{u}$, allowing

to penalize $\Delta\mathbf{u}$ rather than $\mathbf{u}$ directly. However, if any of these additions would improve the landing performance is not certain. The slack variables might be useful if the MAV somehow moves outside the boundaries of the inequality constraints, as the solver might be unhappy with such an event (this was for the *fmincon* solver used in the simulation).

Compensation for sensor, communication and computational delay was done used the internal models of the MAV and the boat. As the times compensated for are relatively small, this approximation is probably not too bad, but a more sophisticated dead-reckoning method could definitely increase the performance to some degree. Dead-reckoning might also be incorporated into the jitter compensation strategy. Here it could be used to estimate the MAV and boat states in a better way (than using the prediction from the MPC), potentially allowing to run new optimizations even without new sensor data from the positioning system.

8.4 Simulation

The scenarios used for simulation was chosen based on the discussion in Section 5.3, but are still somewhat arbitrary. It might have been interesting to run simulations from a larger verity of conditions (starting positions and weather conditions) in order to get a better idea on the landing performance. The current assumptions on the relationship between correlation between wind speed and wave parameters seem reasonable, but it would have been good to explore this topic in more detail.

The simulation time-step could have been set to a lower value for more detail of the "world" in the simulation. However, as most systems were originally modeled as ordinary differential equations (ODEs), a better starting point would probably be to interchange the first order Euler discretization with some more advanced method for numerical integration, for example some variant of the Runge-Kutta method.

As no state estimation for either MAV states or wind was made, the assumption of MAV states and ambient wind being known, with some additional noise, was made. The corresponding estimators are available in the flight stacks that are to be used, so this information is available to the controller. However, in the simulation the variance of the added noise was somewhat arbitrary chosen and for a more realistic result it should have been matched with identified variances from the state estimators used by the flight stacks.

Results

As expected, the results when using the perfect positioning system seem to have the best overall performance. Even if the positioning is perfect is should noted that there is still noise on the MAV and the boat states as well as waves and wind present.

The simulations for the 'perfect' sensors system could be considered an evaluation of how well the landing algorithm is working by its own. For these results it can be noted that lateral displacements seems to be small with good consistency. The vertical displacements are much larger, but this could also be expected since no regard is taken to matching the periodicity of the waves. Thus, if an intersection between the MAV and the boat occurs on a wave ridge or valley, a vertical displacement like this is expected.

The GNSS modules show an expected difference in the results between the two models. From the simulation results it can be concluded that the NEO-M8N is not capable of delivering the required precision, while the ZED-F9P is considerably more reliable. The ZED-F9P has very high lateral precision, while the vertical precision is lower, but not much worse than any other positioning method. This is likely, similarly as for the 'perfect' sensor, a result of the waves affecting the vertical position of the boat, which is problem for every positioning method. The results show that the ZED-F9P GNSS module is a viable candidate.

For the 3D LiDAR case, all models show good results, which is unexpected as the performance of the models vary quite substantially in performance. As previously discussed, this is likely a result of inadequate modeling, false assumptions and over simplification. For the simulation case where the 3D LiDAR has a 30 m detection range, there are landing attempts that fail because of the starting condition, which despite issues with modeling should be considered a valid risk. The MAV never succeeds to enter the volume in which it is visible to the 3D LiDAR. This shows that there is need for a setup phase of the MAV, before it enters the landing phase. A difference in precision is expected between the different 3D LiDAR models, and the lack of such difference in the results is a troubling factor which indicate that the model used is insufficient. Therefore, few conclusions can be drawn about the feasibility of this method.

8.5 Conclusion

The analysis of relative positioning systems indicates that a relative GNSS approach is possible when using the ZED-F9P devices. The simulation and real life implementation test showed an error that was within tolerable range. Used together with an Xbee communication device there seems to be enough robustness for the system to be an implementable alternative. More research and testing is needed to draw conclusions on whether 3D LiDAR is a suitable method since the conclusions drawn in this thesis are based on a large number of assumptions and simplifications. However, relevant research has been and is being conducted specifically on detection of small UAVs which indicate potential in the method. Further, the market for devices can be expected to diversify, and suitable devices may increase in

availability and decrease in price. The small scale ILS system proposed suffered from issues related to precision when performing test measurements. The cause of these issues is likely to be the inherent result of the method itself but further tests is needed to verify this. Other versions using different methods, such as sensing the phase difference between two signals should be considered.

The implementation of the full system for a real world test was not finished, but a small scale implementation test was performed which helped to validate the quality of the full scale simulation and isolated tests.

The proposed landing algorithm seems to be a good starting point for reaching the goal of consistently performing successful landings on a boat during varying weather conditions. However, the optimality of the internal plant model used by the MPC could be questioned, and there are quite a bit of work left to do in order to get ready for real-world flight tests.

In conclusion, given the assumptions that were made in this thesis, it is clear that the GNSS positioning method using two ZED-F9P devices and a Xbee communication device, in combination with the proposed landing strategy is a viable candidate for a first step towards a real-time implementation. As further implementations of positioning methods other than GNSS are investigated, they can benefit from using the same communication method.

8.6 Future Work

Positioning

GNSS Continue to implement a system where the ZED-F9P devices are used. There is a need to evaluate if it is possible for the MAV to use such a large volume helix antenna as the one which were used to perform the tests. Different antenna options could be explored.

3D LiDAR Perform actual testing of 3D LiDARs where the ability to detect the MAV is tested for different orientations and different versions. For example, a new, small, low-cost Velodyne device named Velabit has been proposed, which could be tested and evaluated in the near future.

ILS As the small scale ILS system proposed in this thesis seems to have multiple flaws, other possible versions could be explored. Another version based on the space modulation principles briefly described in Section 3.7 could be evaluated for design. Another suggestion is to make use of the phase difference between a signal sent out by two antennas located at some distance from each other, to measure angular offset from the center line.

Autonomous Landing

On the landing side of the thesis there are a few things that could be explored further in the future. The most obvious thing is of course to make the implementation on a real MAV. For a start, this would include implementation of the landing strategy in ROS (C++ or Python) and make it work within PX4 or Ardupilot. Notice that the Ardupilot-mode corresponding to the PX4 offboard mode is currently not as feature rich when used with planes, and might lack features necessary for being controlled in the way described in the report. Additionally, verification on the computational needs for real-time optimization as well as implementation of decision logic on when and how to abort a landing, would both be necessary steps towards real world usage. Many of the topics covered in the earlier discussion could be interesting to evaluate as well, for example changing the MPC state model or improving the strategy for delay compensation.

Before transferring any system to the MAV for real world tests, the systems could (and probably should) be tested in PX4 or Ardupilot simulation environments. After this, the next step would be to transfer, tune, test and evaluate the low level control systems on the real MAV. Now the needed identification of parameters used in the MPC could be done. Finally, the MPC and high level decision logic could be transferred, tuned, tested and evaluated in real life.

A

Model Parameters

Zagi Airframe Parameters			
Parameter	Value	Parameter	Value
m	1.56	C_{m_q}	-1.399
J_x	0.1147	$C_{m_{\delta_e}}$	-0.3254
J_y	0.0576	C_{Y_0}	0.0
J_z	0.1712	C_{Y_β}	-0.07359
J_{xz}	0.0015	C_{Y_p}	0.0
S_{wing}	0.2589	C_{Y_r}	0.0
b	1.4224	$C_{Y_{\delta_a}}$	0.0
c	0.3302	$C_{Y_{\delta_r}}$	0.19
S_{prop}	0.0314	C_{ℓ_0}	0.0
ρ	1.2682	C_{ℓ_β}	-0.02854
k_{motor}	20.0	C_{ℓ_p}	-0.3209
e	0.9	C_{ℓ_r}	0.03066
C_{L_0}	0.09167	$C_{\ell_{\delta_a}}$	0.1682
C_{L_α}	4.8776	$C_{\ell_{\delta_r}}$	0.0024
C_{L_q}	2.8932	C_{n_0}	0.0
$C_{L_{\delta_e}}$	0.2724	C_{n_β}	0.04
C_{D_0}	0.01631	C_{n_p}	-0.01297
C_{D_α}	0.2108	C_{n_r}	-0.1434
C_{D_p}	0.0254	$C_{n_{\delta_a}}$	-0.00328
C_{D_q}	0.0	$C_{n_{\delta_r}}$	-0.069
$C_{D_{\delta_e}}$	0.3045	C_{prop}	1.0
C_{m_0}	-0.02338	M	50.0
C_{m_α}	-0.5674	ε	0.1592
		α_0	0.4712

Table A.1: The modified parameter-set of the Zagi airframe used in the simulation.

B

Controller Parameters

Roll Controller Parameters	
Parameter	Value
$k_{p,\phi}$	3.5
$k_{i,\phi}$	0.1
$k_{d,\phi}$	-0.3358

Table B.1: Controller gains for the roll PID-controller.

Pitch Controller Parameters	
Parameter	Value
$k_{p,\theta}$	-5
$k_{d,\theta}$	1.0389

Table B.2: Controller gains for the pitch PI-controller.

Course Controller Parameters	
Parameter	Value
$k_{p,\chi}$	5.0226
$k_{i,\chi}$	16.4980

Table B.3: Controller gains for the course PI-controller.

TECS Controller Parameters	
Parameter	Value
k_{p,δ_t}	0.2
k_{i,δ_t}	10
k_{p,δ_e}	0.1
k_{i,δ_e}	1

Table B.4: Parameter values for the energy rate TECS controller.

C

SSRS Custom Designed Airframe Specifications

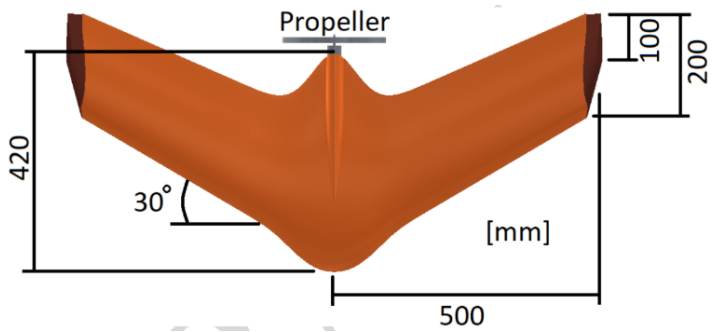


Figure C.1: Top view of airframe.

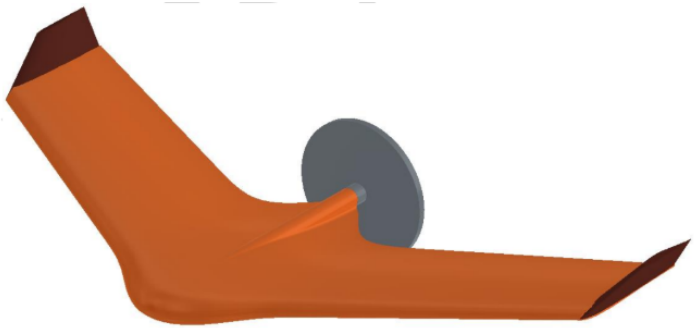


Figure C.2: Side view of airframe.

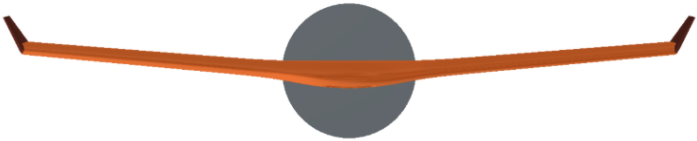


Figure C.3: Front view of airframe.

- Wingspan: 1 m
- Volume: 4413 cm^3
- Mass: 154 g (only EPP-foam, no components added)

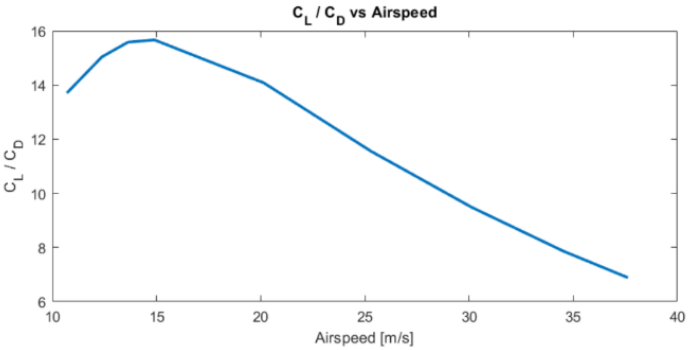


Figure C.4: Glide polar (ratio of lift and drag) at different velocities, for the airframe.

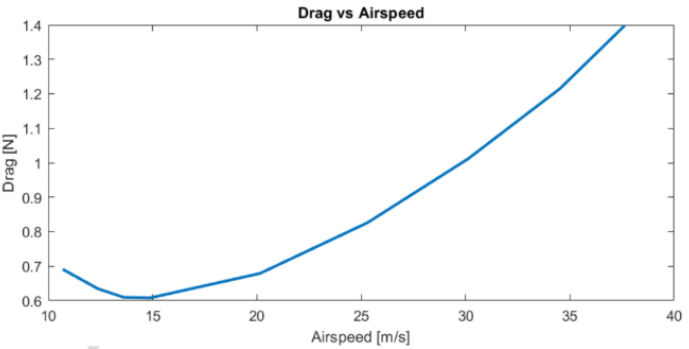


Figure C.5: Drag force for different velocities for the airframe.

Bibliography

Acharya, R. (2014a). *Understanding Satellite Navigation, (Chapter 7)*. Academic Press.

Acharya, R. (2014b). *Understanding Satellite Navigation, (Chapter 8)*. Academic Press.

Alkholidi, A. and K. Altowij (2012). *Effect of Clear Atmospheric Turbulence on Quality of Free Space Optical Communications in Western Asia, p.49 - 65*. InTechOpen.

Argyle, M. E. and R. W. Beard (2016). "Nonlinear total energy control for the longitudinal dynamics of an aircraft". *2016 American Control Conference (ACC)*.

Beard, R. W. and T. W. McLain (2012). *Small Unmanned Aircraft: Theory and Practice*. Princeton University Press.

Errata; Small Unmanned Aircraft: Theory and Practice (2020). URL: `http://uavbook.byu.edu/lib/exe/fetch.php?media=uavbook_supplement.pdf` (visited on 2020-08-18).

Fridén, T. (2020). *Robust Autonomous Landing of Fixed-Wing UAVs in Wind*. Linköping University, Department of Electrical Engineering, Automatic Control.

Gautam, A., P. Sujit, and S. Saripalli (2014). "A survey of autonomous landing techniques for UAVs". *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*.

Ground-Based Navigation - Instrument Landing System (ILS) (2020). URL: `https://www.faa.gov/about/office_org/headquarters_offices/ato/service_units/techops/navservices/gbng/ils/` (visited on 2020-05-12).

Guo, J., H. Zhang, and X.-j. Zhang (2015). "Propagating characteristics of pulsed laser in rain". *International Journal of Antennas and Propagation* **2015**.

Hammer, M., M. Hebel, B. Borgmann, M. Laurenzis, and M. Arens (2018). "Potential of lidar sensors for the detection of uavs". *Laser Radar Technology and Applications XXIII*.

Karaim, M., M. Elsheikh, and A. Noureldin (2018). *GNSS Error Sources*. IntechOpen.

Laird PC2415N datasheet (2020). URL: https://connectivity-staging.s3.us-east-2.amazonaws.com/s3fs-public/2018-10/ANT-DS-PC2415N%5C%200611.pdf (visited on 2020-05-18).

Lecture notes in Optimization Theory. Chapter 4 (2020). URL: https://www.math.uh.edu/~rohop/fall_06/Chapter4.pdf (visited on 2020-05-12).

Mathisen, S., K. Gryte, T. Johansen, and T. Fossen (2016). "Non-linear model predictive control for longitudinal and lateral guidance of a small fixed-wing uav in precision deep stall landing". *AIAA Infotech @ Aerospace*.

McCollum, D. M. (1983). *Evaluation of Instrument Landing System DDM Calibration Accuracies, p.I1-14 - I1-22*. Defense Technical Information Center.

Murray, R. M., Z. Li, S. S. Sastry, and S. S. Sastry (1994). *A mathematical introduction to robotic manipulation*. CRC press, pp. 165–167.

Muskardin, T., G. Balmer, L. Persson, S. Wlach, M. Laiacker, A. Ollero, and K. Kondak (2017). "A novel landing system to increase payload capacity and operational availability of high altitude long endurance uavs". *Journal of Intelligent Robotic Systems* **88**.

Other Global Navigation Satellite Systems (GNSS) (2020). URL: https://www.gps.gov/systems/gnss/ (visited on 2020-05-12).

Ouster OS1 3D LiDAR (2020). URL: https://ouster.com/products/os1-lidar-sensor/ (visited on 2020-05-12).

Ouster OS1 Datasheet (2020). URL: http://data.ouster.io/downloads/OS1-lidar-sensor-datasheet.pdf (visited on 2020-05-12).

Pérez, M. C., D. Gualda, J. de Vicente, J. M. Villadangos, and J. Ureña (2018). "Review of uav positioning in indoor environments and new proposal based on us measurements". *IPIN 2019*.

Persson, L. (2016). *Cooperative Control for Landings Fixed-Wing Unmanned Aerial Vehicle on a Ground Vehicle*. KTH, School of Electrical Engineering.

Persson, L. (2019). *Autonomous and Cooperative Landings Using Model Predictive Control*. KTH, School of Electrical Engineering and Computer Science.

Platanitis, G. and S. Shkarayev (2005). "Integration of an autopilot for a micro air vehicle". *Infotech @ Aerospace*.

PX4 - Autopilot User Guide (2020). URL: https://docs.px4.io/master/en/ (visited on 2020-08-31).

SMHI - Nederbörd (2020). URL: https://www.smhi.se/data/meteorologi/nederbord (visited on 2020-05-18).

Smith, M. (1988). *Introduction to Antennas, p.73 - 77*. Macmillan Education LTD.

SSRS rescue fleet (2020). URL: https://www.sjoraddning.se/vara-batar (visited on 2020-05-12).

Total Energy Control for Longitudinal Autopilot (2020). URL: https://uavbook.byu.edu/lib/exe/fetch.php?media=shared:tecs_autopilot.pdf (visited on 2020-05-12).

u-Blox NEO-M8 datasheet (2020). URL: https://www.u-blox.com/sites/default/files/NEO-M8-FW3_DataSheet_%5C%28UBX-15031086%5C%29.pdf (visited on 2020-05-18).

u-Blox ZED-F9P datasheet (2020). URL: https://www.u-blox.com/sites/default/files/ZED-F9P_DataSheet_%5C%28UBX-17051259%5C%29.pdf (visited on 2020-05-18).

Valero Beltrá, D. (2018). *Design and Mechatronic Integration of a Drone Launcher*. Chalmers University of Technology, Department of Electrical Engineering.

Van Valkenburg, M. E. (2002). *Reference Data for Engineers: Radio, Electronics, Computers and Communications*. Elsevier Inc.

Velodyne Product Guide (2020). URL: https://velodynelidar.com/downloads/ (visited on 2020-05-12).

VHF Nav (2020). URL: https://www.opticfox.com/2a4x2/vor-fun.htm (visited on 2020-05-04).

Voigt, J. and A. Alkaysi (2020). *Autopilot for a Personal Watercraft*. Lunds Tekniska Högskola, Department of Automatic Control.

Wang, Y. and S. Boyd (2010). "Fast model predictive control using online optimization". *IEEE Transactions on Control Systems Technology* **18**.

Wikimedia Commons, M. B. via (2007). *A basic working principle of model predictive control*. File: LambdaPlaques.jpg. URL: https://commons.wikimedia.org/wiki/File:MPC_scheme_basic.svg (visited on 2020-05-27).

Wojtanowski, J., M. Zygmunt, M. Kaszczuk, Z. Mierczyk, and M. Muzal (2014). "Comparison of 905 nm and 1550 nm semiconductor laser rangefinders' performance deterioration due to adverse environmental conditions". *Opto-Electronics Review* **22**.

Zagi website (2020). URL: https://zagi.com/product/thl/ (visited on 2020-06-10).

Lund University
Department of Automatic Control
Box 118
SE-221 00 Lund Sweden

Document name
MASTER'S THESIS

Date of issue
October 2020

Document Number
TFRT-6104

Author(s)
Olle Fagerström Hedbrant
Martin Gemborn Nilsson

Supervisor
Marcus Greiff, Dept. of Automatic Control, Lund University, Sweden
Anders Robertsson, Dept. of Automatic Control, Lund University, Sweden
Karl-Erik Årzén, Dept. of Automatic Control, Lund University, Sweden (examiner)

Title and subtitle

Robust Relative Positioning & Autonomous Landing of a Flying Wing MAV for Sea Rescue Applications

Abstract

In this thesis, a study is made into relative positioning and control systems for flying wing Micro Aerial Vehicles (MAVs) intended to be used by the Swedish Sea Rescue Society (SSRS) for the purpose of providing assistance during sea rescue missions. The particular scenario of interest is robust relative positioning to be used for autonomous landing of a MAV on a boat at the end of a rescue mission. Three positioning methods, Global Navigation Satellite System (GNSS), 3D LiDAR and a scaled down version of an Instrument Landing System (ILS) are explored and evaluated through physical testing, modeling and simulation. In addition, a control system based on Model Predictive Control (MPC) is proposed for autonomous landing of a flying-wing MAV on a moving boat. A full simulation of the target scenario, including positioning systems, control system and environmental factors is made, from which conclusions about the feasibility of successfully erforming a landing for the combined positioning system and landing algorithm are drawn. The results of the thesis indicate, given the proposed landing strategy, that a high performance GNSS would be sufficient to complete this type of landing consistently. The 3D LiDAR solution also shows potential, but is based on a number of assumptions and simplifications in modeling. The small scale ILS method proposed suffered from precision issues that make it unfit for the target scenario.

Keywords

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title
0280-5316

ISBN

Language
English

Number of pages
1-111

Recipient's notes

Security classification