Towards Vision-Based Autonomous Cross-Country Soaring for UAVs

Martin Stolle

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par : *l'Institut Supérieur de l'Aéronautique et de l'Espace (ISAE)*

Présentée et soutenue le ***03/04/2017*** **par :**

Martin Tobias STOLLE

Towards Vision-Based Autonomous Cross-Country Soaring for UAVs

JURY

Mme Caroline Chanel	Ingénieur Chercheuse	Membre du Jury
M. Carsten Döll	Ingénieur Chercheur	Co-Directeur de thèse
Mme Isabelle Fantoni	Directrice de Recherche	Rapporteur
M. Yann Le Gorrec	Professeur d'Université	Président du Jury
M. Robert Luckner	Professeur d'Université	Rapporteur
Mme Yoko Watanabe	Ingénieur Chercheuse	Co-Directrice de thèse

École doctorale et spécialité :
EDSYS : Automatique 4200046
Unité de Recherche :
ONERA - Department of Information Processing and Systems
Directeurs de Thèse :
M. Gilles Ferrères (HDR), M. Carsten Döll et *Mme Yoko Watanabe*
Rapporteurs :
Mme Isabelle Fantoni et *M. Robert Luckner*

Abstract

Small fixed-wing Unmanned Aerial Vehicles (UAVs) provide utility to research, military, and industrial sectors at comparably reasonable cost, but still suffer from both limited operational ranges and payload capacities. Thermal soaring flight for UAVs offers a significant potential to reduce the energy consumption. However, without remote sensing of updrafts, a glider UAV can only benefit from an updraft when hitting it by chance.

In this thesis, a new framework for autonomous cross-country soaring is elaborated, enabling a glider UAV to visually localize sub-cumulus thermal updrafts and to efficiently gain energy from them.

Relying on the Unscented Kalman Filter (UKF), a monocular vision-based method is established, for remotely estimating sub-cumulus updraft parameters. Its capability of providing convergent and consistent state estimates is assessed by Monte Carlo Simulations. Model uncertainties, image processing noise, and poor observer trajectories can degrade the estimated updraft parameters. Therefore, a second focus of this thesis is the design of a robust probabilistic path planner for map-based autonomous cross-country soaring. The proposed path planner balances between the flight time and the risk of an outlanding by taking into account the estimation uncertainties in the decision making process. The suggested updraft estimation and path planning algorithms are jointly assessed in a 6 Degrees of Freedom (DOF) simulator, highlighting significant performance improvements with respect to state of the art approaches in autonomous cross-country soaring while it is also shown that the path planner is implementable on a low-cost computer platform.

But those who trust in the LORD will find new strength. They will soar high on wings like eagles. They will run and not grow weary. They will walk and not faint.

-Isaiah 40:31 (NIV)

To My Parents - With Love and Gratitude, and above all, To the almighty God.

Acknowledgements

This thesis was not a solo effort. I am grateful for every individual who supported me.
First I would like to thank Carsten Döll for providing every kind of help from the very early stage of my project. Thanks a lot Carsten for having been supportive with the scholarship and the valuable exchange we had talking about flight control and soaring techniques.
I would also like to thank Yoko Watanabe for the countless hours she dedicated to my work. Thank you for having shared with me your knowledge and for having helped with improving and concretizing many of the ideas of this work.
I am also grateful towards Caroline Chanel, having spent so much time with me talking about path planning concepts. I am also grateful for your encouragement.
My committee, Robert Luckner, Isabelle Fantoni, and Yann Le Gorrec, thank you for your interest in my work, and for the enriching comments and the advice.
I would also like to thank the Multiplex Modellsport GmbH & Co.KG for having provided fully equipped Cularis RC gliders that were valuable for collecting flight data and video records.
To my friend Jan Bolting, thank you a lot for the past years. I'm still thinking of the times where we started working together in the TU Braunschweig UAV lab, especially the UAV competitions in Florida, and of course our excursions to Hossegor.
To Leandro Lustosa, I am very happy having met you at the countless ISAE lunch sessions and grateful for having shared an office with you during the last period of my work.
To Timothy van Oorschot, I had never such a nice roommate who turned out to be a divine cook, a balanced and grate character, and a good friend. Thank you also for the unforgettable Atlantic Ocean weekends. I must say these Cap Ferret days were part of my best times in France.
To François Bachoc, thank you for having been a supportive roommate and for your countless advices.
To my brother and sister Simon and Charlotte. Thank you for the phone calls, the reunions,

the holidays, the dinners, the conversations. I am grateful for being your brother.
To my parents: I am grateful for your support on all levels, your trust, for having tought me curiosity, for having helped me on my way, and for your devotion. I'd never have made my way without you!
Finally and most importantly I thank God. I am grateful for Your guidance and unconditional love. I am really looking forward to everything You have lined up for me next.

Contents

List of Figures

List of Tables

List of Symbols and Acronyms

If not stated otherwise, scalars are noted as non-bold, vectors as bold, and matrices as bold and/or non-bold uppercase.

Symbols

Alphanumeric

$\mathbf{b}_L, \mathbf{b}_R$	Left/Right bearing to the cloud feature points, m
b	Wingspan, m
w	Climb rate of the glider, m/s
$\mathbf{d}$	Cloud size evolution vector, $\left[\mathrm{m}, \mathrm{m/s}, \mathrm{m/s}^2\right]$
d_u	Updraft spacing, km
f	Focal length, m
$\mathbf{p}$	Vehicle position vector (Inertial frame), m
$\mathbf{p}_c$	Cloud position vector, m
$\mathbf{p}_u$	Updraft position vector (inertial frame), m
q_d	Horizontal distance between the vehicle and the destination position, m
q_r	Horizontal distance between the outlanding site and the destination position, m
S_x, S_y	Conversion factors from meter to pixel, px/m
t_v	Updraft vanish time, s
V_a	Airspeed, m/s
$V_a^\star$	Final glide airspeed, m/s

V_g	Ground speed, m/s
V_{MC}	MacCready airspeed, m/s
v_{si}	Sink rate, m/s
w_p	Core updraft velocity, m/s
$w_{p,z}$	Core updraft velocity averaged over altitude, m/s
$w^\star$	Convective velocity scale, m/s
w_u	Updraft velocity, m/s
$\bar{w}_{u,rz}$	Average updraft velocity of all updrafts with respect to both the updraft radius and the altitude, m/s
z	Altitude, m
z_i	Convective mixing layer thickness, m
z_{cb}	Cloud base altitude, m

Greek letters

δ	Angle between the heading to the cloud base center and the feature points of the cloud, rad
γ	Flight path angle, rad
κ	Coefficient relating the updraft velocity to the cloud diameter amplitude, 1/s
ν	Zero mean Gaussian measurement noise
Ψ	Heading, rad
Ψ	Penalty for outlandings, s
ρ_u	Updraft density, $1/\text{km}^2$
Σ	Diagonal matrix with standard deviations
τ_c	Lifespan of a cloud, s
ϵ	Normalized estimation error squared, [-]
ξ	Root mean squared error

Acronyms

AHRS	Attitude Heading Reference System
ALOFT	Autonomous Locator of Thermals
AVL	Athena Vortex Lattice

CCL	Cumulus Condensation Level
COTS	Commercials Off-The-Shelf
CTG	Cost-To-Go
DALR	Dry Adiabatic Lapse Rate
DOF	Degrees of Freedom
EA	Eigenstructure Assignment
EKF	Extended Kalman Filter
FAI	Fédération Aéronautique Internationale
GPS	Global Positioning System
HSV	Hue Saturation Value
IMU	Inertial Measurement Unit
LES	Large Eddy Simulation
LIDAR	Light Detection And Ranging
MBCCS	Map-Based Cross-Country Soaring
MC	Monte Carlo
MCS	Monte Carlo Simulation
MDP	Markov Decision Process
NASA	National Aeronautics and Space Administration
NEES	Normalised Estimation Error Squared
NRL	Naval Research Laboratory
PDF	Probability Density Function

RGB	Red Green Blue
RMSE	Root Mean Square Error
SAS	Stability Augmentation System
SSP	Stochastic Shortest Path
SSPMDP	Stochastic Shortest Path Markov Decision Process
UAV	Unmanned Aerial Vehicle
UKF	Unscented Kalman Filter
UT	Unscented Transform
VBACCS	Vision-Based Autonomous Cross-Country Soaring

Indices

$\mathbf{d}$	Refers to the cloud size dynamics vector $\mathbf{d}$
c	Refers to the cloud
xy	Refers to the horizontal, i.e. xy components of a 3D xyz vector

Chapter 1

Introduction

Vision-based autonomous soaring is an unexplored and promising way to extend both the endurance and range of fixed-wing UAVs. This subject poses challenges in atmospheric modeling, flight control, image processing, non-linear estimation and path planning. Among these, the underlying work focuses on the estimation and path planning parts and results in two main contributions:
The first one refers to the vision-based remote estimation of sub-cumulus updrafts. It provides an estimator for remotely sensing updraft parameters that are necessary to perform efficient autonomous cross-country soaring flight.
The second contribution is a robust, map-based path planning algorithm for autonomous cross-country soaring. It balances between the risk of outlandings and the minimization of flight time by considering uncertainties of updraft map parameters.

1.1. Background and Motivation

In the past decades, UAVs have proven to be of great utility, facilitating a variety of tasks while simultaneously reducing operating costs. Many applications such as border patrolling or pipeline monitoring require the UAV to cross significant distances, or simply to stay aloft for long periods of time, e.g. remote sensing or atmospheric research. Both these requirements however, challenge especially small and electrically powered vehicles. This is primarily due to one important aerodynamic performance metric, the maximum lift-to-drag ratio. High lift-to-drag ratios call for high wing aspect ratios, conflicting with the small size of these vehicles. Additionally, the small carriage capacity is often utilized in favor of the payload, rather than

the battery.

While it has become common practice to interrupt the mission in order to change batteries, the development of batteries with higher energy densities might be a long-term solution. Other solutions aim at weight reduction by using advanced materials such as carbon fiber, or include the design of aerodynamically more efficient airframes. Even the development of flight control laws for UAV formation flight is considered to reduce energy consumption [5], mimicking the behavior of migratory birds.

In comparison, both range and endurance of a UAV can also be extended by making use of external energy - provided by "mother nature". For example, efforts were conducted to equip UAVs with solar panels for an in-flight battery recharge [6, 7]. Another option is to use external energy in order to perform unpropelled flight. Unpropelled flight, widely known as soaring, is complementary to the aforementioned approaches. Compared to the development of high performance batteries, the use of solar panels or advanced materials, soaring solely necessitates suited algorithms and in some cases additional sensors. Most generally, soaring flight comprises the two fields of dynamic and static soaring.

Dynamic Soaring Horizontal wind gradients are used to increase the kinetic energy in dynamic soaring, a method that is commonly adopted by maritime birds such as Albatrosses. It enables them to stay aloft for hours and to travel significant distances without taking a rest. Figure 1.1 illustrates the flight trajectory of an Albatross during dynamic soaring. Nature has motivated researchers in bringing this capacity to UAVs [1, 8]. Although dynamic soaring for UAVs sounds promising, a successful realization of this method relies on sensors with an accuracy that enables flight in extreme ground proximity. For maritime applications, the sensor has to be accurate enough to allow a reliable "below"-surface operation [9].

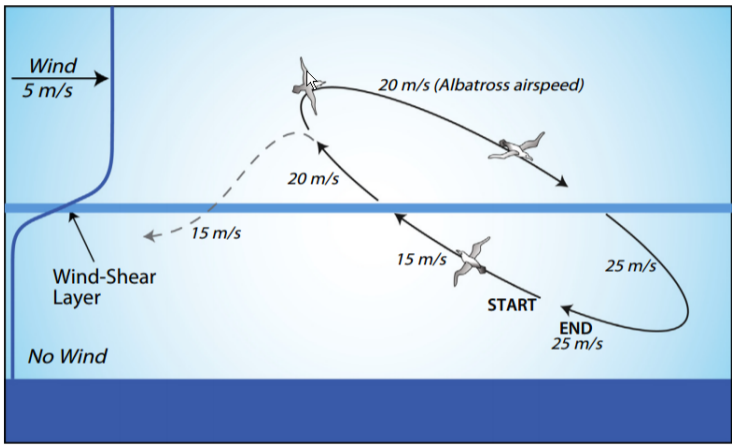


Figure 1.1: Flight trajectory of an Albatross bird in dynamic soaring [1]

Static Soaring Contrary to dynamic soaring, static soaring necessitates the presence of vertical winds. As part of static soaring, slope soaring, also known as ridge soaring, relies on vertical winds that are generated when horizontal winds hit an obstacle. The procedure is illustrated in Figure 1.2. These objects are typically ridges or steep coasts. In urban environments, obstacles can also be buildings. Numerous studies were presented for generating slope soaring trajectories for UAVs (e.g. [10, 11, 12]).

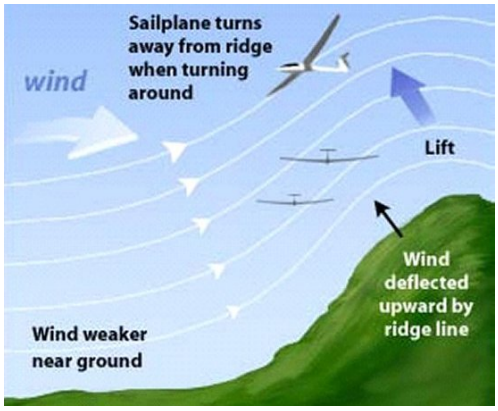


Figure 1.2: Illustration of ridge soaring (`http://www.aerospace.org`)

Vertical winds can also be generated by solar radiation, heating the ground and creating columns of rising air. These columns are referred to as thermal updrafts. Thermal soaring for UAVs was first investigated in [13]. As reported by the author of [14], UAVs with a nominal endurance of two hours can increase the flight time by 12 hours in the summer and 8 hours in the winter respectively - only relying on thermal updrafts. This significant potential has motivated researchers to work on autonomous thermal soaring. The first successful realization of an autonomous soaring system was conducted by the NASA in 2007 [2] using a small-scale glider, as illustrated in Figure 1.3. Its successful flight test campaign has since inspired a lot of succeeding research, for example path and motion planning for autonomous gliders [15, 16], updraft estimation [17, 18], and cooperative soaring [19].

Figure 1.3: NASA Autonomous Soaring UAV [2]

Compared to other energy sources such as updrafts caused by ridges, thermal updrafts repeatedly emerge over large geographical regions. This fact is exploited in cross-country soaring where gliders/birds fly beyond the gliding distance from the initial take-off point and perform waypoint navigation. The first and sole realization of an autonomous cross-country soaring flight was published by Dan J. Edwards et al., the authors of [3].

Figure 1.4: ALOFT autonomous glider [3]

The work led to the participation in a cross-country soaring challenge for remotely piloted gliders. Figure 1.4 illustrates the glider plane used by Edwards. The UAV could perform a fully autonomous cross-country soaring flight over a distance of approximately $50\,\text{km}$. The corresponding flight trajectory is depicted in Figure 1.5.

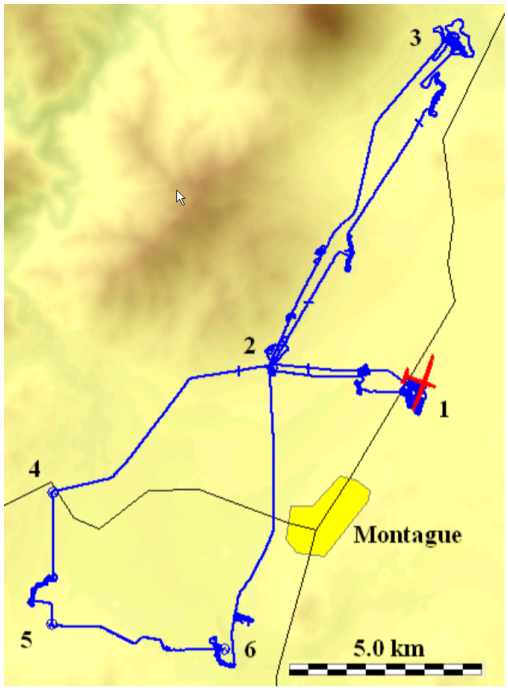


Figure 1.5: Flight log of the ALOFT glider on day one of the competition [3]

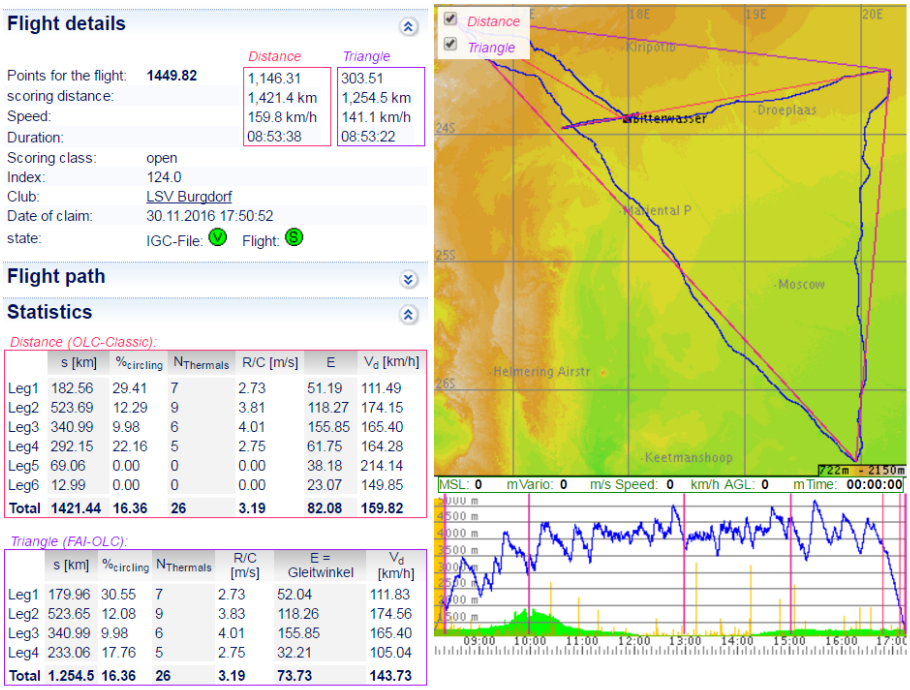
With no a priori information on thermal locations in the far environment, the UAV flight path

was defined as the direct line between two consecutive waypoints. The vehicle's flight control mode was set to thermal centering mode, when encountering strong enough thermals. Thermals were detected by Inertial Measurement Unit (IMU), Global Positioning System (GPS) and airdata measurements. These quantities can be transformed in order to monitor the vehicle's total energy rate. While the 50 km flight was a giant step towards autonomous cross-country soaring, the application of the system is restricted to weather conditions with high updraft densities. Additionally, the lack of future updraft knowledge results in reactive and therefore inefficient flight trajectories. As a consequence, to this day, autonomous cross-country soaring still bares a significant risk of mission failure and therefore finds only limited application.
In contrast, human glider pilots rely on their vision when performing soaring. More specifically, pilots remotely estimate parameters of sub-cumulus updrafts by observing cumulus clouds that lie in the vehicle's environment [20, 21] (see Figure 1.6).

Figure 1.6: Cumulus soaring conditions in Pokweni, Namibia (`http://www.gliding.ch`)

Doing so, glider pilots can fly distances of more than 1000 km within a few hours only. This evidence is illustrated in Figure 1.7. It shows a flight log of a long distance cross-country soaring flight, performed over the desert of Namibia. As illustrated, the average ground speed of the vehicle is about 140 km/h over a cumulated distance of 1400 km. With that said, the sky provides important information for efficient soaring flight, that to this date is not yet exploited by soaring UAVs.

Therefore, the aim of this work is to contribute to the applicability of autonomous cross-country soaring by exploiting visual information.



Flight details

		Distance	Triangle
Points for the flight:	**1449.82**	1,146.31	303.51
scoring distance:		1,421.4 km	1,254.5 km
Speed:		159.8 km/h	141.1 km/h
Duration:		08:53:38	08:53:22
Scoring class:	open		
Index:	124.0		
Club:	LSV Burgdorf		
Date of claim:	30.11.2016 17:50:52		
state:	IGC-File: V Flight: S		

Flight path

Statistics

Distance (OLC-Classic):

	s [km]	%circling	NThermals	R/C [m/s]	E	Vd [km/h]
Leg1	182.56	29.41	7	2.73	51.19	111.49
Leg2	523.69	12.29	9	3.81	118.27	174.15
Leg3	340.99	9.98	6	4.01	155.85	165.40
Leg4	292.15	22.16	5	2.75	61.75	164.28
Leg5	69.06	0.00	0	0.00	38.18	214.14
Leg6	12.99	0.00	0	0.00	23.07	149.85
Total	**1421.44**	**16.36**	**26**	**3.19**	**82.08**	**159.82**

Triangle (FAI-OLC):

	s [km]	%circling	NThermals	R/C [m/s]	E = Gleitwinkel	Vd [km/h]
Leg1	179.96	30.55	7	2.73	52.04	111.83
Leg2	523.65	12.08	9	3.83	118.26	174.56
Leg3	340.99	9.98	6	4.01	155.85	165.40
Leg4	233.06	17.76	5	2.75	32.21	105.04
Total	**1.254.5**	**16.36**	**26**	**3.19**	**73.73**	**143.73**

Figure 1.7: Flight log of a 1400 km FAI triangle over Namibia (data and figure taken from `http://www.onlinecontest.org`)

1.2. A Framework for Autonomous Cross-Country Soaring

When solar radiation hits the ground, the air in the vicinity of the earth's surface is heated and therefore expands. The expanding, warm, and moist air rises, and forms a column of lifting air. While rising, the air cools down until it reaches the Cumulus Condensation Level (CCL) where cumulus clouds form. As long as the updraft remains active, its capping cloud grows. Once the updraft has vanished, the cloud size begins to decay. The cumulus cloud thus not only indicates the position of the updraft, but its geometry even provides information on the activity of the updraft [20].

In order to stay aloft by benefiting from these energy sources, autonomous soaring systems necessitate several components. Of these, the most fundamental components are both energy-state estimation and thermal centering control. Given the rate of change of the vehicle's energy, bank angle control commands can be generated, for centering the glider within the updraft and thereby increasing its potential energy level.

A more complex soaring task is cross-country soaring, as illustrated in Figure 1.8. Beyond the

elementary task of staying aloft, the vehicle is additionally challenged to cross a distance in order to reach a target destination. To fulfill this task, a series of updrafts has to be visited, before the destination lies in the gliding range of the vehicle. For increased reliability, a more

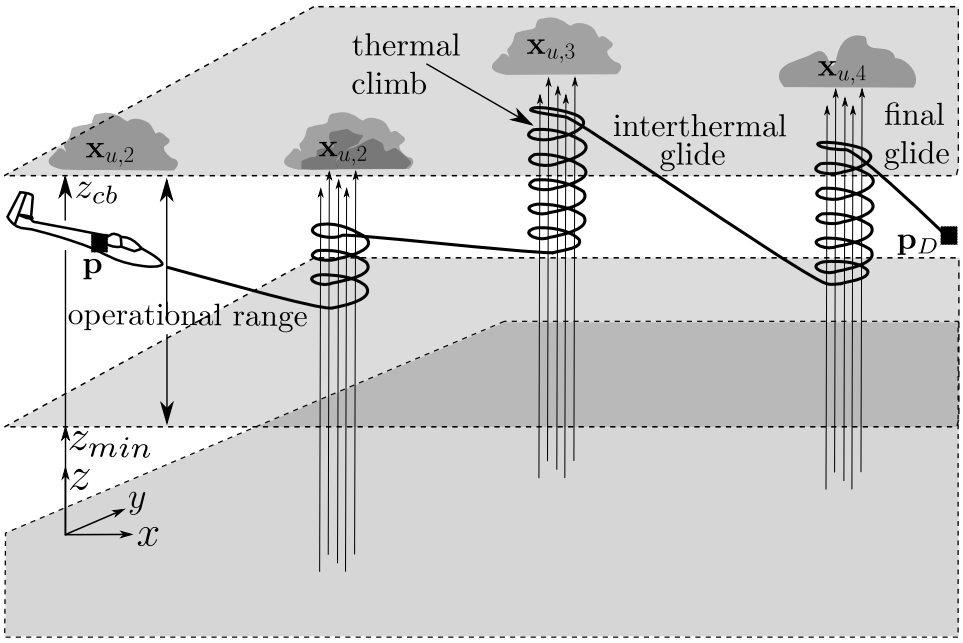


Figure 1.8: Cross-country flight in cumulus soaring conditions

advanced system is necessitated enabling soaring flight performances that are comparable to those of a self-propelled UAV. Such a system not only relies on energy-state estimation and thermal centering control, but requires two more components. When aiming to prevent both reactive and "luck"-dependent trajectories, remote updraft estimation functionality is essential. Given a map with estimated updrafts, a suited path planner is additionally required, for guiding the vehicle to its destination while minimizing/maximizing given performance criteria such as flight time.

To this date, such an advanced system does not exist. Therefore, a framework as illustrated in Figure 1.9 is proposed in this thesis. This vision-based framework includes the mentioned system components. More specifically, both a Red Green Blue (RGB) camera and an image processing algorithm form a vision unit that provides vision measurements of cumulus clouds. Vision, position, and attitude measurements, as provided by the Attitude Heading Reference System (AHRS) and the GPS are fused for remotely estimating positions $\mathbf{p}_u$, vanish times t_v and strengths $w_{u,rz}$ of the updrafts. The generated updraft map serves as input to a path planner that computes a sequence of updrafts-to-visit, necessitated to reach the destination, as given by the mission. High-level control inputs are transmitted to a flight guidance system that generates bank angles and airspeed commands which are tracked by a set of dedicated

flight control laws.

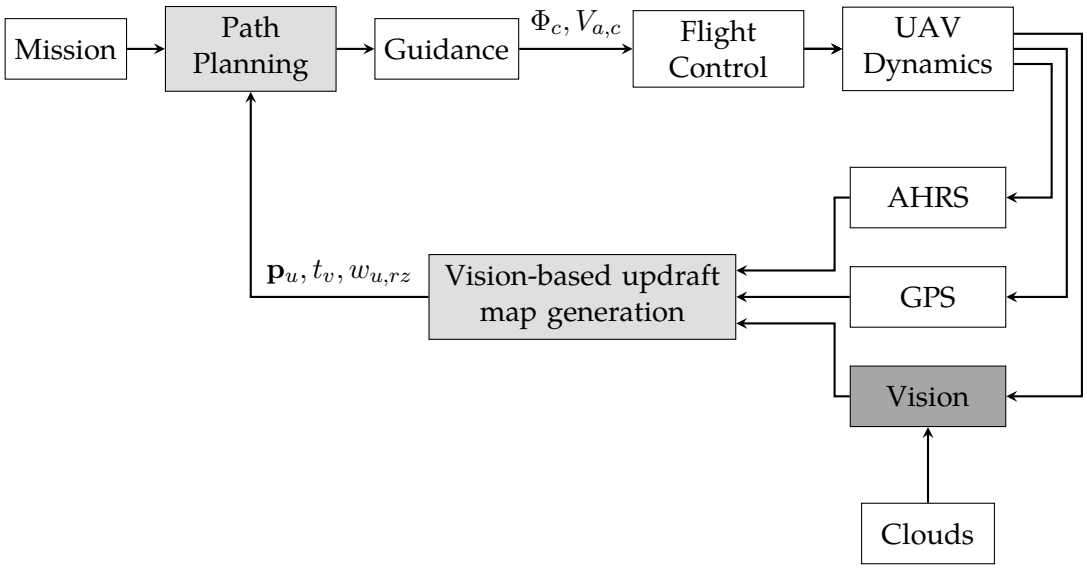


Figure 1.9: A vision-based framework for reliable autonomous cross-country soaring

1.3. Related Work

The primary components that enable Vision-Based Autonomous Cross-Country Soaring (VBACCS) are updraft estimation, path planning, and thermal centering control. The following section provides an overview of the state of the art on these fields.

1.3.1. Atmospheric Modeling

The design of autonomous soaring algorithms depends on atmospheric models that describe the meteorological phenomena a glider encounters during soaring flight. Evidently, methods like Large Eddy Simulation (LES), commonly used by meteorologists [22], provide a rather realistic description of the atmosphere. As these models however tend to be computationally costly, they do not provide a practical solution for the design of soaring algorithms. A measurement based updraft model, specifically dedicated to the design of soaring algorithms and widely-used in the community, is provided in [23]. A rather simple model that approximates updrafts as a Gaussian lift distribution can be found in [13]. Both these models are limited in the way that they do not consider horizontal winds. Therefore, Lawrance [24] has made efforts to provide models that include the interaction between horizontal wind and updrafts.
For vision-based autonomous soaring, not only updrafts but also clouds have to be modeled.

Ideally, these models are backed by measurements of local pressure, humidity, and wind-flow, as well as cloud width measurements. Therefore, meteorologists have equipped small UAVs with sensors, in order to collect in-cloud measurements at high resolution [25]. While in-cloud data collection is feasible, it remains challenging to collect cloud width measurements over time. Therefore, suited image processing algorithms as well as special observer flight trajectories are required, when aiming to rely on UAV systems. Cloud width measurements can also be collected by an observer that is located on the ground. This was done in [4]. To the author's knowledge, this is the only publication of cloud width measurement data. In this work, these measurements are utilized to derive a simplified updraft cloud model which enables the design of vision-based autonomous soaring algorithms.

1.3.2. Updraft Estimation

The updraft estimation problem can be grouped in two fields: *In updraft* and *remote updraft* estimation.

In updraft estimation deals with the estimation of the vehicle's vertical velocity. Therefore, the vehicle's energy-state has to be determined. The authors of [2] propose a filter for estimating the vehicle's energy-state and its derivative from static and total pressure measurements while climbing within the updraft. The position, radius, and strength of the thermals are estimated using a history of total energy rate and aircraft positions, stored in a first-in-first-out queue. In [26], a Savitzky-Golay filter is suggested for estimating the total energy, rate of change of total energy, and the second derivative of the total energy of the vehicle by making use of airspeed and altitude measurements. A recursive in-updraft estimation method, based on the Unscented Kalman Filter (UKF) is presented in [27], aiming to reduce the computational requirements for the updraft position estimation. Yet another method is established by the author of [28], where a system for simultaneous exploration and exploitation of static wind fields is suggested, relying on model-free Gaussian Process regression.

The energy state estimation method from [2] has successfully been flight tested and demonstrated how endurance can be increased by relying on energy provided by thermal updrafts. More specifically, the method was flight tested by the NASA in 17 flights during which a total of 23 thermal updrafts were encountered and soaring flights of up to 40 min were performed. Similar soaring flight durations of about 40 min are reported by the authors of [29], which rely on the same energy state estimator to generate thermal centering control inputs.

As stated by the authors of [3, 24], soaring performances can greatly be improved with in-flight

remote sensing of updrafts. Researchers have therefore focused on cooperative [30, 31], and distributed autonomous soaring [32], where data from multiple autonomous soaring aircraft is combined in order to gain information on remote updraft positions. While this technique sounds promising at first glance, an updraft might have vanished once the glider that received the information arrives. To this date, no literature can be found on active remote updraft estimation, i.e updraft estimation that is conducted on-board a UAV.
Vision-based *remote updraft* estimation, as studied in this thesis, deals with the estimation of the updraft position, i.e. the position of the updraft-capping cloud, and also the updraft strength and vanish time. This is a computer vision/image processing and non-linear estimation problem. Both these fields are discussed in the following.

1.3.3. Image Processing

While the focus of this thesis does not lie on image processing, a Master thesis project was conducted in parallel for this purpose [33]. For the sake of completeness, some important contributions related to the identification of clouds in images are yet summarized.
Applying computer vision techniques for cloud identification and classification is a challenging task. Clouds generally differ in size, color, texture, and geometry. Ground based or near ground based imagery, i.e. images that are taken from a perspective where the camera is located underneath the cloud contain blue pixels which originate from the sky as well as "white" pixels originating from clouds. In order to identify geometrical features of a cloud, cloud pixels have thus to be separated from non-cloud pixels. Then, cloud contours/geometrical properties can be identified.
Unlike the blue sky, clouds scatter the blue and red visible light more equally. Therefore, red/blue ratio filters have been employed for separating cloud from blue sky pixels. This approach was followed by the authors of [34, 35] with a constant filter threshold. Others [36] adopt this technique with a varying filter threshold, thus adapting the filter performance to different lightnings. More recently, cloud pixel separation techniques based on the Hue Saturation Value (HSV) color model have also been suggested. Compared to the RGB model, it provides more robustness to changes in the lightning [37]. The authors of [37] provide a method for performing cloud contour detection making use of the Suzuki Abe algorithm, as introduced in [38].

1.3.4. Vision-Based Target Localization

Given vision, position, and attitude measurements, the cloud position and its geometric properties can be estimated. The position estimation process with a single camera is a monocular vision-based navigation problem. The idea is to take imagery data of the target from different vehicle locations and attitudes. Thereby, the target location can be computed by triangulation. This target localization problem, also known as bearings-only target localization, is well-known and has its origins in the maritime sector. More specifically, in sonar applications for submarines.

Despite the prominence of this problem, the function that relates measurements to the target position is highly non-linear, which can be challenging when dealing with observer-trajectories of low observability.

The Extended Kalman Filter (EKF) is a widely used filter to cope with these nonlinearities. The author of [39] for example relies on the EKF for estimating the position of stationary targets. Others [40] adopt the EKF for vision-based estimation of moving obstacles occuring in the flight path of a UAV. However, the EKF is sensitive to initialization and measurement errors and prone to both covariance collapse and other filter instabilities, as was reported in [41].

For reducing the fundamental drawbacks of the EKF, researchers [42] have suggested the so-called Unscented Kalman Filter (UKF) where the covariance is propagated through sampling, circumventing errors introduced by linearization.

Others [39] have focused on the design of trajectories that maximize the observability of the target, thus mitigating the instability of the EKF.

Yet other researchers [43] have focused on increasing the measurement equation linearity by using inverse depth parametrization, thereby also minimizing the likelihood of filter collapses during initialization with large depth uncertainties.

1.3.5. Flight Control Law Design

Control laws for autonomous thermal soaring generally deal with the calculation of bank angle inputs that maximize the climb rate of the UAV glider, when circling within an updraft. In [31], commanded bank angles are calculated by taking the vehicle's specific energy rate and its derivative as control feedback signal. In [44], a theoretical stability analysis of this controller is provided. Flight test results of the same controller are presented in [29]. A slightly different control approach is adopted by the authors of [2], where the bank angle command is not only

function of the vehicle's energy rate, but also of the estimated updraft center.
Concerning the glider's longitudinal motion, airspeed controllers are required to damp both the short period and the phugoid mode. The authors of [15] apply robust adaptive longitudinal control for a soaring UAV in the presence of large model uncertainties and actuator saturations.

1.3.6. Guidance and Path Planning

Path planning and guidance for autonomous cross-country soaring deals with the calculation of airspeed settings and the selection of the updrafts to approach in order to reach the target. The probably most famous publication concerning the airspeed calculation in cross-country soaring was written by physicist Paul B. MacCready, Jr., author of [45]. In this publication, an optimal airspeed setting is suggested for approaching an updraft and regaining altitude in minimum time. The airspeed setting is calculated using the glide ratio of the vehicle and the expected vertical velocity of a designated updraft. A consequence of this research is the invention of the speed-to-fly ring, a basic equipment in contemporary glider planes. Other researchers have extended this idea [46] to address the problem of optimal airspeed calculations under uncertain atmospheric conditions.
Most generally, the goal of an autonomous cross country soaring flight is to reach a remote target in minimum time with the target lying beyond the initial gliding range. In [47], this problem is solved by using classical shortest path algorithms such as the Floyd-Warshall Algorithm. The authors of [48] provide a heuristic based planning solution for large updraft maps. In both these contributions, the authors assume the availability of an updraft map that contains perfect position, climb rate, and vanish time information over the entire planning horizon. The few contributions that concentrate on path planning also underline the open problem of remote updraft estimation which is vital for path planning.

1.4. Summary of Contributions

The main contributions of this work are summarized below:

Framework for Vision-Based Autonomous Cross-Country Soaring The first contribution of this thesis is the establishment of a vision-based flight guidance and navigation system for

autonomous cross-country soaring. The concept of using RGB vision for autonomous cross-country soaring has not been studied prior to this work, and thereby provides a novelty on the field of autonomous soaring.

Remote Updraft Estimator The second contribution of this work is a vision-based estimator for the remote estimation of sub-cumulus updrafts. The estimator not only provides the position, but also the vertical wind velocity and vanish time of updrafts. It is furthermore shown, that the estimator provides consistent state estimates which allows for an integration of estimation uncertainties in the decision-making process.

Robust Path Planner As a third contribution, this work establishes a robust path planner for vision-based autonomous cross-country soaring which takes into account the estimation uncertainty of an imperfect updraft map. The path planner is shown to be implementable on a low-cost glider UAV, using Commercials Off-The-Shelf (COTS) electronic devices.

Simulation Evaluation For illustrating the benefits of the suggested methods, an extensive Monte Carlo (MC) simulation is conducted in a 6 DOF flight simulator with realistic glider dynamics. Performance metrics are established in order to allow a comparative study with the state of the art cross-country soaring system.

1.5. Reader's Guide

The remainder of this thesis is organized as follows:
Chapter 2 describes the design of the remote updraft estimator. This includes the description of the atmospheric and sensor models as well as the filter formulation. Monte Carlo (MC) simulations are conducted to verify the functionality of the proposed estimator in terms of convergence and consistency.
Chapter 3 explains the development of a robust path planning method for map-based autonomous cross country soaring. Therefore, a simple glider model is introduced, and a guidance law for airspeed selection is developed. The mathematical formulation of the planner is explained and simulation results are shown to illustrate the functionality of the path planner.
In Chapter 4, computer simulation results of the combined vision-based flight guidance and navigation system are presented and compared to prior systems.
Chapter 5 summarizes the results of this thesis and provides an outlook on future work.

Chapter 2

Vision-Based Updraft Map Generation

This chapter presents a method for the vision-based, on-board, and real-time estimation of sub-cumulus updrafts. This includes the estimation of the updraft position, velocity, and vanish time, and their respective uncertainties.

Among others, the estimator relies on atmospheric and camera models, and a description of the observer dynamics. In this work, these models are selected to be as simple as possible while capturing the most significant aspects of the estimation problem. The focus in this dissertation is on general features of the remote updraft estimation and not on the particular choice of models. The estimator is designed to be flexible in the way that more complex models can easily be integrated while it remains capable of providing consistent state estimates.

This chapter is structured as follows:

In Section 2.1, the estimation problem is introduced. Section 2.2 presents an atmospheric model for updrafts and clouds which is required for the estimator design. Section 2.3 introduces the sensor models. Section 2.4 presents the estimator design for the updraft map generation. In Section 2.5, the performance of the updraft map generation algorithm is assessed and discussed by showing computer simulation results. Section 2.6 provides a brief summary of the estimator design.

2.1. Problem Statement

Consider a glider plane, deployed in an area with thermal updrafts that are capped by cumulus clouds, as illustrated in Figure 2.1. The glider is equipped with a forward pointing camera and image processing capacity, enabling a detection of cumulus clouds in the image. Here, the camera is assumed to be mounted at the vehicle's center of gravity. As a consequence, the position of the camera coincides with the position of the glider which is denoted by $\mathbf{p}$. The camera's field of view is assumed not to be occluded by parts of the glider's airframe. If not explicitly indicated, all position vectors are expressed in the inertial frame.

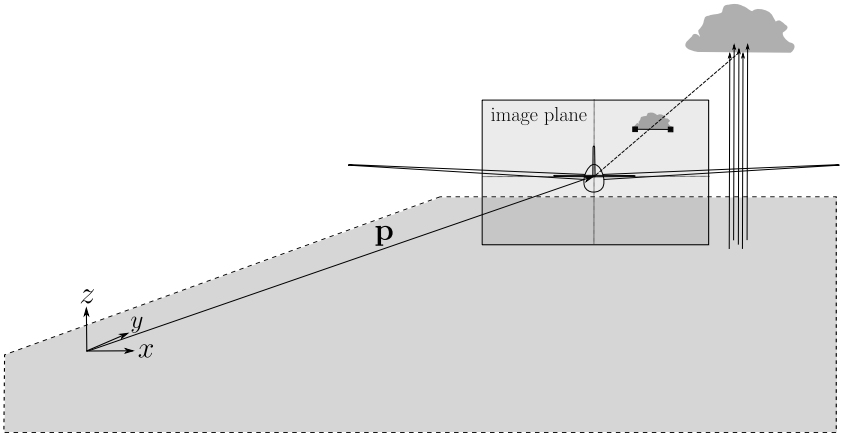


Figure 2.1: Setup between glider, camera and cloud

As discussed by the authors of [20, 21], certain features of a cumulus cloud contain information on the updraft strength and the life cycle of the updraft. In this context, the cloud size plays a major role. More specifically, there is a positive correlation between the cloud size and the updraft strength [20]. In addition, the evolution of the cloud's size indicates whether the updraft is still active or not. This is because a cumulus only grows while it is fed by an updraft and decays otherwise.

The updraft strength and vanish time, as well as the updraft position are crucial values for the decision-making in cross-country soaring as they enable the pilot/autopilot to evaluate the mission-relevant energy that can be gained from the updraft. Rather than randomly hitting updrafts, the autopilot can then generate more efficient flight trajectories and guide the vehicle to the target with less risk of mission failure [3].

In the following, the state of an updraft is denoted $\mathbf{x}_u$ and includes the seeked parameters, i.e.

its position $\mathbf{p}_u$, its vanish time t_v, and its velocity $w_{u,rz}$

$$\mathbf{x}_u = \begin{bmatrix} \mathbf{p}_u \\ t_v \\ w_{u,rz} \end{bmatrix}, \tag{2.1}$$

where the updraft position is noted $\mathbf{p}_u = \begin{bmatrix} p_{u_x} \, p_{u_y} \, p_{u_z} \end{bmatrix}^T$. The term $w_{u,rz}$ specifies the average updraft velocity of an updraft column with bell-shaped velocity distribution where the average refers to both the updraft radius r and the altitude z.

Given the individual updraft states $\mathbf{x}_{u,i}$, the state of an updraft map $\mathbf{X}_u$ is defined to contain all n perceived updrafts such that

$$\mathbf{X}_u = \begin{bmatrix} \mathbf{x}_{u,1} \\ \mathbf{x}_{u,2} \\ \vdots \\ \mathbf{x}_{u,n} \end{bmatrix}. \tag{2.2}$$

The generation of such a map is an unexplored research subject and a complicated task. This is for various reasons.

For a small and low-cost realization of an estimator, vision is the only sensor that can be employed to actively identify and estimate remote updrafts. A Light Detection And Ranging (LIDAR) based estimation would conflict with the size, weight, and cost limitations. To determine the position of cumulus clouds, UAV state estimates are combined with information gathered by vision sensors. This nonlinear, bearings-only estimation problem is sensitive to the relative motion between the observed cloud and the vehicle. Additionally, in autonomous cross-country soaring, the flight trajectories of the glider are chosen to optimize the cross-country soaring velocity, but not the estimation observability. This fact complicates the position estimation since clouds might be poorly observable. Furthermore, the low-cost realization of such a system cannot rely on a 360 degree camera and thus the camera features a limited field of view. Hence, vision measurements for tracked clouds are not constantly available which degrades the estimation performance. Additionally, the benefits of stereo vision can hardly be exploited due to the small ratio between the vehicle's size, i.e. the baseline, and the distance to the clouds which typically is around 3-5 km. Vision measurements and an estimate of the cloud's position can be transformed into the current diameter of the cloud. Observing the evolution of this parameter allows to gather information on updraft strength and lifespan [20]. Imperfections of the image processing algorithms however not only degrade the estimate, but also increase

the initialization time which reduces the timespan available for the UAV to benefit from an updraft.

2.2. Atmospheric Model

On days with favorable meteorological soaring conditions, updrafts create so-called cumulus clouds, when rising parcels of warm and humid air reach the Cumulus Condensation Level (CCL). As discussed in [20], the clouds persist after the thermal has dissipated, and vanish progressively. Thus, the presence of a cumulus not necessarily indicates a thermal beneath its base. Therefore, it is important for human glider pilots to study features of clouds to differentiate clouds that can be beneficial from those that are not.
That being said, the design of a vision-based remote updraft estimator relies on an atmospheric model which describes the relation between the updraft state and cloud characteristics.
However, to this date, there does not exist a simple and suited model for this purpose in the literature. Therefore, in this work, a simple model is derived which makes the design of an estimator possible by capturing the most significant relations between the updraft state and clouds. The design of more precise atmospheric models that still balance complexity and performance is identified as future work and will be discussed in Chapter 5.
The purpose of the following section is thus the description of a model which relates the states of an updraft $\mathbf{x}_u$ to the state of its related cloud $\mathbf{x}_c$

$$\mathbf{x}_u = f(\mathbf{x}_c). \tag{2.3}$$

2.2.1. Thermal Updrafts

Convective mixing of the planetary boundary layer results in vertically oriented wind columns, referred to as thermal updrafts [49]. In this work, we rely on the model from [23] to describe the vertical velocity profile of an updraft.

Velocity profile: According to Allen's updraft model [23], the velocity profile of an updraft at a given altitude features a bell shape, i.e. the updraft velocity has a peak at the core and decreases with the radius. At a given altitude z, the averaged updraft velocity of all updrafts

$\bar{w}_{u,r}$ depends on the altitude z according to

$$\bar{w}_{u,r}(z) = \left(\frac{z}{z_i}\right)^{\frac{1}{3}} \left(1 - 1.1\frac{z}{z_i}\right) w^{\star}. \tag{2.4}$$

where $w^{\star}$ stands for the vertical velocity scale and z_i is the convective mixing layer thickness. Notice that the index r specifies the average over the radius, i.e. from $-r$ to r, and the bar stands for the average of all updrafts.

Average updraft velocity over altitude: Building upon Equation (2.4), the average updraft velocity $\bar{w}_{u,rz}$ is introduced as the average of the velocity $\bar{w}_{u,r}$ over a given altitude range, i.e. between some minimum altitude z_{min} and the cloud base altitude z_{cb}. It is obtained by integrating Equation (2.4) over the altitude z and dividing by the altitude range $z_{cb} - z_{min}$

$$\bar{w}_{u,rz} = \frac{\int_{z_{min}}^{z_{cb}} \bar{w}_{u,r}(z)dz}{z_{cb} - z_{min}} = \frac{w^{\star}}{z_{cb} - z_{min}} \left[\frac{3}{4}\frac{z^{\frac{4}{3}}}{z_i^{\frac{1}{3}}} - \frac{33}{70}\frac{z^{\frac{7}{3}}}{z_i^{\frac{4}{3}}}\right]_{z_{min}}^{z_{cb}}. \tag{2.5}$$

Note that in this notation the index rz means averaged over both the radius and the altitude. To model variations for individual updrafts, we can scale $\bar{w}_{u,rz}$ with a perturbation gain c

$$w_{u,rz,i} = c_i \bar{w}_{u,rz}. \tag{2.6}$$

Vanish time: As discussed by the authors of [50], the updraft velocity of a thermal barely varies with time. However, updrafts feature a limited duration. In order to model these characteristics, the perturbation gain c is defined to be constant at some positive value during the active phase of the thermal, and drops to zero at the time instance t_v where the thermal vanishes

$$c = \begin{cases} \text{constant} > 0 & \text{, if } t_0 \leq t < t_v \\ 0 & \text{, otherwise.} \end{cases} \tag{2.7}$$

Position: The position of an updraft is defined as the updraft's horizontal center position at the cloud base altitude z_{cb}

$$\mathbf{p}_u = \begin{bmatrix} p_{u_x} \\ p_{u_y} \\ z_{cb} \end{bmatrix}. \tag{2.8}$$

In this work, we refer to weather scenarios with no horizontal wind. Therefore, the horizontal position derivative of the updraft is null. As a consequence, updrafts can be modeled as non-tilted, non-bended and torsion free.

The cloud base altitude slowly rises during the morning and descends with the nocturnal inversion [49]. Updrafts would thus exhibit a slow vertical movement. This phenomenon can however be neglected in the updraft model, since the duration of an updraft ($\sim 20\,\text{min}$ according to [23, 21]) is rather short, compared to the rise/sink time of the cloud base level. That being said, the derivative of the updraft position vector is noted as

$$\dot{\mathbf{p}}_u = \mathbf{0}_{3\times 1}. \tag{2.9}$$

Remark on the notation: In the remainder of this work, a matrix $\mathbf{0}_{i\times j}$ defines a null matrix of dimension $i \times j$, and a matrix $\mathbf{I}_{i\times j}$ can be a pseudo identity matrix, such that

$$\mathbf{I}\,(i,j) = \begin{cases} 1 & \text{, if } i = j \\ 0 & \text{, otherwise.} \end{cases} \tag{2.10}$$

To give an example, $\mathbf{I}_{2\times 3}$ is defined as

$$\mathbf{I}_{2\times 3} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}. \tag{2.11}$$

2.2.2. Clouds

The state of a cloud $\mathbf{x}_c$ can be represented by two uncorrelated vectors. These are the cloud position vector $\mathbf{p}_c$, and a vector $\mathbf{d}$ that contains parameters which describe the evolution of the cloud size w.r.t. time

$$\mathbf{x}_c = \begin{bmatrix} \mathbf{p}_c \\ \mathbf{d} \end{bmatrix}. \tag{2.12}$$

Cloud position: With the definition of the updraft position from Equation (2.8), both cloud and updraft position coincide. Therefore, the following holds

$$\mathbf{p}_c = \mathbf{p}_u \text{ and } \dot{\mathbf{p}}_c = \dot{\mathbf{p}}_u. \tag{2.13}$$

As shown in [21] and further illustrated in Figure 2.2, the cloud base altitude z_{cb} is the line intersection of the Dry Adiabatic Lapse Rate (DALR) and the dew point. According to the authors of [51], the cloud base altitude z_{cb} can be calculated according to

$$z_{cb} = \frac{T_0 - T_{d,0}}{\frac{\partial T}{\partial z} - \frac{\partial T_d}{\partial z}} \approx 125 \left[\frac{\mathrm{m}}{K}\right] (T_0 - T_{d,0}), \tag{2.14}$$

where T and T_d are the temperature of the air and the dew point respectively. The parameters T_0 and $T_{d,0}$ are the respective temperatures at the ground, as shown in Figure 2.2. Only tiny variations of the temperature differences between the air and the dew point can be found when flying over regions with constant geological and orographic structure. These variations are considered to be non-significant in the mission operation zone. Thereby, the cloud base altitude z_{cb} in the observed region can be considered as uniform for all clouds. This phenomenon is illustrated in Figure 2.2, and additionally depicted in Figure 2.3. Furthermore, this phenomenon can be exploited in the design of the cloud position estimator, as will be elaborated in Section 2.4.3.

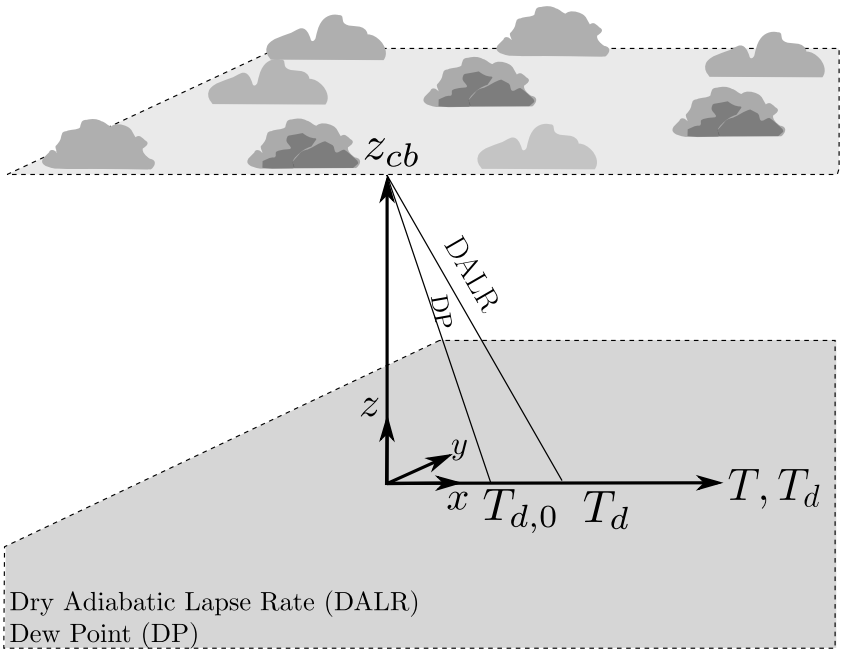


Figure 2.2: Uniform cloud based altitude

Cloud Size: In this work, we refer to cumulus humilis clouds, a type of clouds which is generated by thermal updrafts, appearing in fair weather conditions and covering up to 25% of the sky [21]. As explained earlier, the size properties of these clouds contain information on both the updraft strength and vanish time. According to the author of [52], one characteristic

Figure 2.3: Field of cumulus humilis clouds

of this cloud type is the constant width-to-height ratio. Consequently, both height and width of a fair-weather cumulus contain redundant size information on the updraft parameters and only one has to be observed in order to retrieve the seeked parameters. Selecting the width rather than the height as measurement has two advantages. When observing clouds that are behind one another w.r.t. the camera, the front cloud can occlude the height of the cloud that lies behind, but not its width, when the observer is supposed to be located below the cloud base altitude. Since we want to extract the size property of the cloud from images that are recorded below the cloud base altitude, it is hence more promising to refer to the width.
Furthermore, cumulus clouds feature rather flat bases which facilitates the detection of this characteristic in an image.
As discussed by the authors of [4], the base of a cumulus cloud can be approximated by a circle with diameter d. The diameter grows while the updraft is active ($t < t_v$), and decays once it starts dissipating ($t > t_v$). Inputs to the process that describes the cloud width over time also include physical quantities such as the local humidity or air pressure. However, for a vision-only remote updraft estimation, measurements of these meteorological parameters under the remote cloud are not available. Thus, the on-board model has to be stated as a function of time only, as schematically illustrated in Figure 2.4.

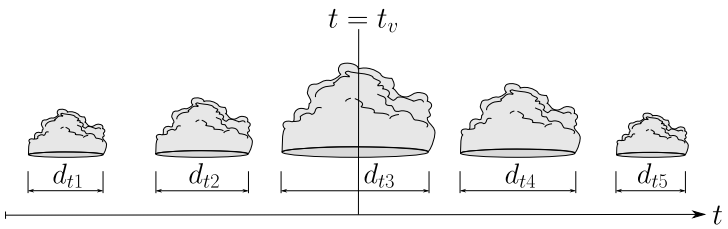


Figure 2.4: Lifecycle of a cumulus

The authors of [4] further provide time series measurements of cumulus cloud widths. To the author's best knowledge, this is the only published source of empirical cloud width evolution data, and as such a valuable source for a measurement-backed identification of the cloud width evolution function. The width measurements for four clouds are illustrated in Figure 2.5. Note that the clouds were not necessarily tracked over the whole lifespan which explains the width variations at time zero.

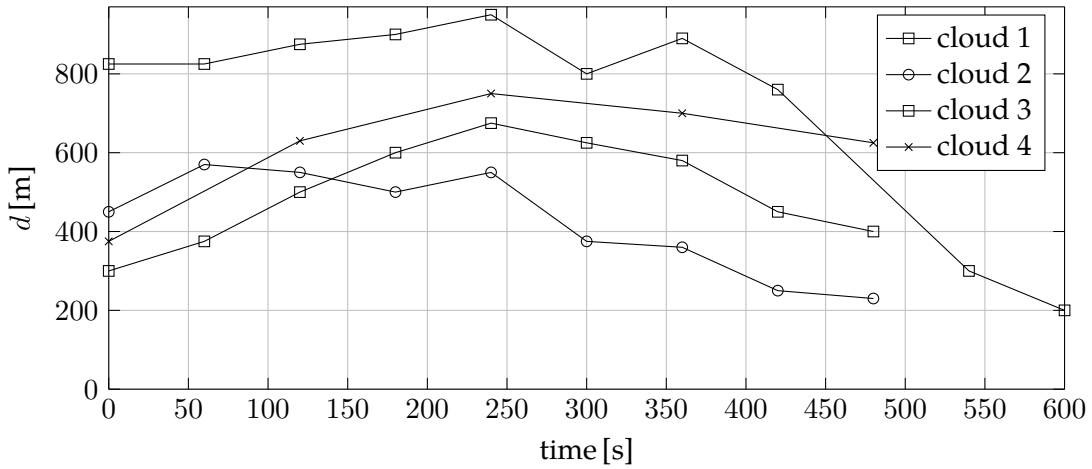


Figure 2.5: Cloud diameter evolution from [4]

In this work, a simple polynomial function is used for modeling the cloud size dynamics

$$d(t) = e_1 t^2 + e_2 t + e_3 \text{ with } e_1 < 0. \tag{2.15}$$

Figure 2.6 illustrates both the measurements and parabola-fitted cloud widths for cloud 1 from Figure 2.5. Note that the error between both fit and measurements remains rather small (up to approximately 10%) for all sample points. However, due to both the small number of measurement samples and the lack of information on the measurement uncertainty it is hard to quantify the accuracy of this model. Therefore, the modeling error of Equation (2.15) cannot be reliably quantified due to the lack of statistically relevant empirical data sets. This explains why model errors w.r.t. this function are not considered in this work. The collection of further measurement data and the quantification of model errors constitutes an open problem for future work in this domain. That being said, the second component of the cloud's state, i.e. the vector $\mathbf{d}$ is defined as

$$\mathbf{d} = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix}. \tag{2.16}$$

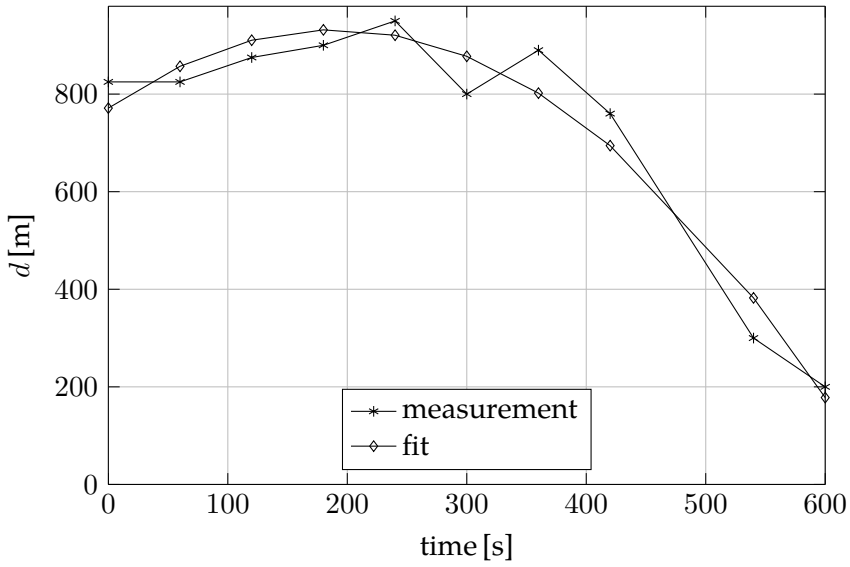


Figure 2.6: Fitted and measured cloud width evolutions

Inserting Equation (2.16) into Equation (2.12) yields

$$\mathbf{x}_c = \begin{bmatrix} \mathbf{p}_c & \mathbf{d} \end{bmatrix}^T = \begin{bmatrix} p_{c_x} & p_{c_y} & z_{cb} & e_1 & e_2 & e_3 \end{bmatrix}^T. \tag{2.17}$$

2.2.3. Relation between Updraft Lifespan, Strength, and Cumulus Clouds

A decaying cloud indicates that its driving updraft has vanished. Hence, the vanish time t_v corresponds to the time when the cloud's size begins to decrease, as shown in Figure 2.4. Given Equation (2.16), the vanish time is defined by the abscissa of the cloud evolution parabola's vertex

$$t_v = -\frac{e_2}{2e_1}. \tag{2.18}$$

Stronger updrafts, i.e. updrafts that feature a higher updraft velocity generally result in a more significant cloud size. This phenomenon can be modeled by a linear relation between the cloud size amplitude d_{max} and the updraft velocity

$$w_{u,i} = \kappa\, d_{max,i}, \tag{2.19}$$

where the parameter d_{max} is related to the coefficients of the parabola according to

$$d_{max} = e_3 - \frac{{e_2}^2}{4\, e_1} > 0 \tag{2.20}$$

and κ is some constant defined by local meteorological weather conditions and supposed to be known.

The updraft-cloud relation is then summarized as

$$\mathbf{x}_u = \begin{bmatrix} \mathbf{p}_u^T & t_v & w \end{bmatrix}^T = f(\mathbf{x}_c) = \begin{bmatrix} \mathbf{p}_c^T & -\frac{e_2}{2\,e_1} & \kappa \left(e_3 - \frac{{e_2}^2}{4\,e_1} \right) \end{bmatrix}^T . \tag{2.21}$$

2.3. Sensor Modeling

2.3.1. Camera Model

A pinhole camera model, as illustrated in Figure 2.7, is used to conduct the perspective projection from objects onto the image plane.

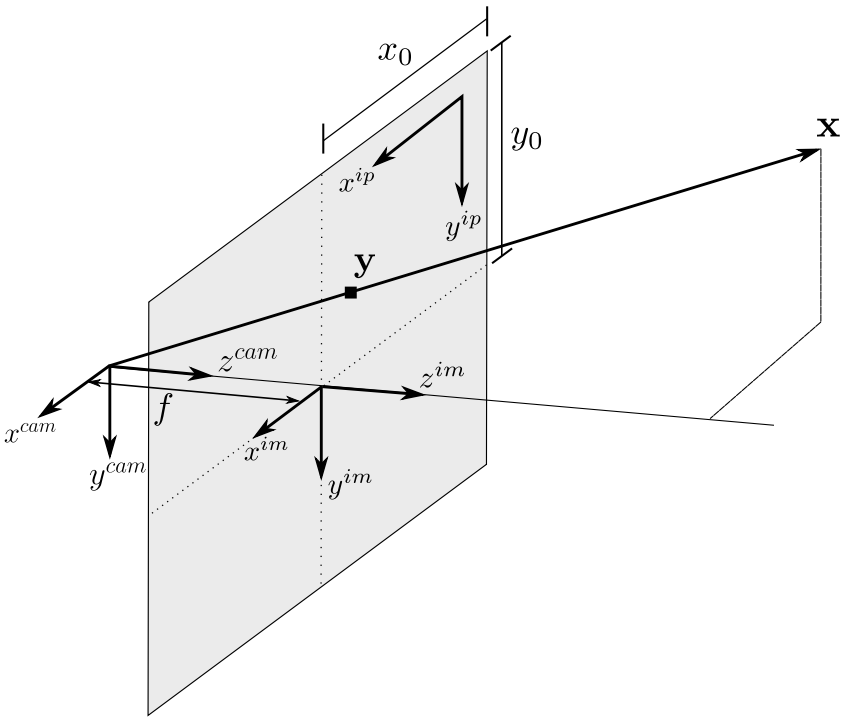


Figure 2.7: Pinhole camera model

Note that the z axis of the camera frame is aligned with the optical axis and the superscript cam stands for camera. The joint transformation and unit conversion of a vector $\mathbf{x}^{cam} \in \mathbb{R}^3$

with unit meters into a vector $\mathbf{x}^{ip} \in \mathbb{R}^2$ with unit pixels in the image frame is given by

$$\mathbf{x}^{ip} = \mathbf{I}_{2\times 3} \left(\frac{1}{x_z^{cam}} \underbrace{\begin{bmatrix} f_x & 0 & x_0 \\ 0 & f_y & y_0 \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{C}} \mathbf{x}^{cam} \right), \tag{2.22}$$

where $\mathbf{C}$ specifies the camera calibration matrix, x_0 and y_0 are the offsets from the upper-left hand corner to the image center in pixels and f_x and f_y depend on both the focal length f and conversion factors S_x and S_y respectively

$$\begin{aligned} f_x &= f S_x, \\ f_y &= f S_y. \end{aligned} \tag{2.23}$$

The conversion factors are the inverse size of a pixel in the respective axis and for a quadratic pixel, $S_x = S_y$.

2.3.2. Image Processing

The vision-based updraft estimation builds upon the fact that one can determine the updraft position, strength, and lifespan by observing cumulus clouds. As a minimum visual information, the two feature points $\mathbf{y}_L$ and $\mathbf{y}_R$ which are the endpoints of the cloud's base in the image are required. These points are shown in Figure 2.8.

Figure 2.8: Feature points

They contain the information on the current cloud size, given an estimate of the cloud position $\mathbf{p}_c$. Recent advances in image processing techniques [37] demonstrate the feasibility of detecting cumulus cloud contours in near ground imagery. Therefore, in parallel to this PhD research, a master thesis project [33] was launched, aiming to study image processing methods

for retrieving the required measurements. In order to design the image processing algorithms, a low-cost RGB camera was mounted on a remotely piloted airplane, and video data was collected on a day with fair weather cumulus soaring conditions, creating a rich data set of images. The crucial image processing steps that resulted from the master thesis are:

1. Separation of cloud and clear sky pixels,
2. Cloud segmentation,
3. Cloud contour detection,
4. Baseline detection.

A binary histogram back-projection (see Figure 2.9b) is applied onto the original image (see Figure 2.9a) separating cloud from blue sky pixels in the sky part of the image, i.e. above the horizon line. Note that the horizon line can easily be reconstructed from the vehicle's known attitude which helps reducing the computational effort by applying the image processing only in the sky part of the image. In a second step, a watershed transformation is applied on the binary image to distinguish clouds from each other, i.e. to retrieve single clouds (Figure 2.9c). For identified clouds, contour detection is applied in order to identify the polygon describing the contour of the cloud, as shown in Figure 2.9d. Finally, the cloud baseline is obtained by calculating the intersection points of the cloud contours with a line that is parallel to the horizon (Figure 2.9e).
In what follows, an image processing algorithm that provides these two feature points is considered to be available and a vision measurement $\mathbf{y}_{LR} \in \mathbb{R}^4$ is introduced as

$$\mathbf{y}_{LR} = \begin{bmatrix} \mathbf{y}_L \\ \mathbf{y}_R \end{bmatrix} = h(\mathbf{x}_c) + \boldsymbol{\nu}_{\mathbf{y}_{LR}}, \tag{2.24}$$

where the vector $\boldsymbol{\nu}_{\mathbf{y}_{LR}}$ stands for zero-mean Gaussian measurement noise

$$\boldsymbol{\nu}_{\mathbf{y}_{LR}} \sim \mathcal{N}\left(0, \Sigma^2_{\mathbf{y}_{LR}}\right) \tag{2.25}$$

and the measurement equation h in Equation (2.24) relates the cloud states to measurements. The diagonal matrix $\Sigma^2_{\mathbf{y}_{LR}}$ in Equation (2.25) specifies the measurement noise covariance of the vision measurement.
Remark on the notation: All noise terms introduced hereinafter follow the definition of Equation (2.25) and are noted $\boldsymbol{\nu}_{\mathbf{x}}$ and $\boldsymbol{\nu}_{\mathbf{y}}$ respectively. Subscripts $\mathbf{x}$ and $\mathbf{y}$ distinguish between pro-

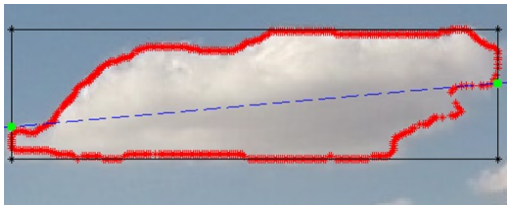
cess and measurement. The related sub-subscripts, (e.g. $\nu_{\mathbf{y}_{LR}}$) denote the corresponding signal.

(a) Original image

(b) Binary back-projection filtering

(c) Watershed

(d) Contour detection

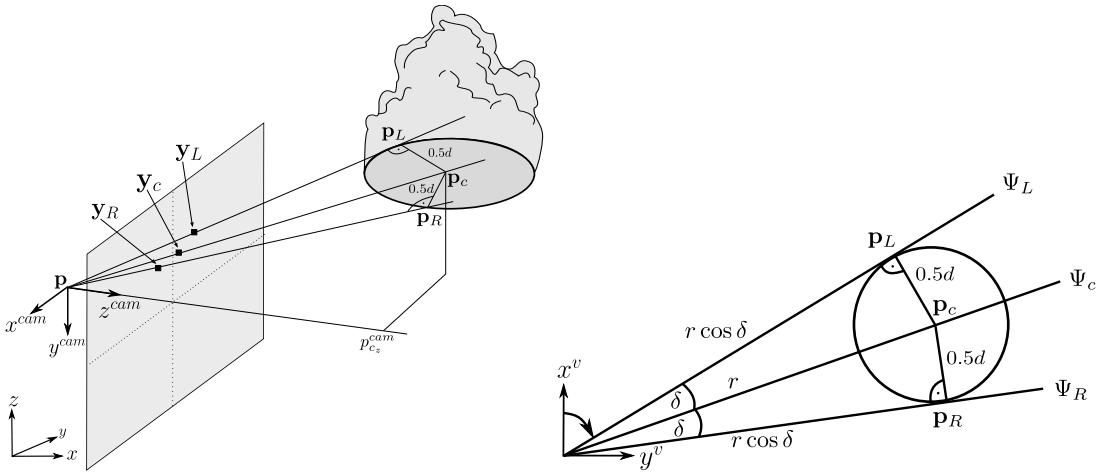


(e) Cloud baseline

Figure 2.9: Image processing steps

2.3.3. Center Point Sensor

As will be discussed in Section 2.4.2, we separate the updraft position estimation from the estimation of the vanish time and the updraft strength. For the cloud position estimation, we rely on vision measurements $\mathbf{y}_c \in \mathbb{R}^2$ of the cloud's center position $\mathbf{p}_c$. This measurement can be reconstructed from the vision measurement $\mathbf{y}_{LR}$ such that $\mathbf{y}_c = g(\mathbf{y}_{LR})$. The geometry between the camera and the three points $\mathbf{p}_L, \mathbf{p}_R$, and $\mathbf{p}_c$ is illustrated in Figure 2.10.



(a) Spatial constellation (b) xy-plane

Figure 2.10: Geometry between $\mathbf{y}_c$ and $\mathbf{y}_{LR}$

Note that the aircraft position vector $\mathbf{p}$ and the two points $\mathbf{p}_L$ and $\mathbf{p}_R$ span a plane, and $\mathbf{p}_c$ not necessarily lies on that plane. Also note that Ψ stands for the heading to the respective point and recall that d is the current diameter of the cloud.

In order to obtain the vision measurement $\mathbf{y}_c$, we start by decomposing $\mathbf{y}_{LR}$ as follows

$$\begin{bmatrix} \mathbf{y}_L \\ 1 \end{bmatrix} = \mathbf{C}\frac{\mathbf{p}_L^{cam}}{p_{L_z}^{cam}} = \frac{1}{p_{L_z}^{cam}}\mathbf{C}\mathbf{R}_{cam\leftarrow v}\left(\mathbf{p}_L - \mathbf{p}\right), \tag{2.26}$$

$$\begin{bmatrix} \mathbf{y}_R \\ 1 \end{bmatrix} = \mathbf{C}\frac{\mathbf{p}_R^{cam}}{p_{R_z}^{cam}} = \frac{1}{p_{R_z}^{cam}}\mathbf{C}\mathbf{R}_{cam\leftarrow v}\left(\mathbf{p}_R - \mathbf{p}\right), \tag{2.27}$$

where the matrix $\mathbf{R}_{cam\leftarrow v}$ transforms a vector from the vehicle frame (noted v) to the camera frame. The vehicle frame is defined as an inertial frame with its origin at the vehicle's position. We can rewrite Equations (2.26) and (2.27), and solve for the positions of the two outer points

in the vehicle frame as function of the bearings $\mathbf{b}_L^v$ and $\mathbf{b}_R^v$

$$\mathbf{p}_L^v = \mathbf{p}_L - \mathbf{p} = p_{L_z}^{cam} \underbrace{\mathbf{R}_{cam\leftarrow v}^{-1}\mathbf{C}^{-1}\begin{bmatrix}\mathbf{y}_L\\1\end{bmatrix}}_{\mathbf{b}_L^v} = \begin{bmatrix} r\cos\delta\begin{bmatrix}\cos\Psi_L\\ \sin\Psi_L\end{bmatrix}\\ \Delta p_z\end{bmatrix}, \tag{2.28}$$

$$\mathbf{p}_R^v = \mathbf{p}_R - \mathbf{p} = p_{R_z}^{cam} \underbrace{\mathbf{R}_{cam\leftarrow v}^{-1}\mathbf{C}^{-1}\begin{bmatrix}\mathbf{y}_R\\1\end{bmatrix}}_{\mathbf{b}_R^v} = \begin{bmatrix} r\cos\delta\begin{bmatrix}\cos\Psi_R\\ \sin\Psi_R\end{bmatrix}\\ \Delta p_z\end{bmatrix}. \tag{2.29}$$

It is important to note that the bearings are transformed vision measurements, i.e. available quantities. Also note that Δp_z stands for the vertical distance between the glider and the cloud. The two bearings $\mathbf{b}_L^v$ and $\mathbf{b}_R^v$, i.e. the bearings from the glider to the feature points in the vehicle frame can be normalized as follows

$$\hat{\mathbf{b}}_L^v = \begin{bmatrix}\cos\Psi_L\\ \sin\Psi_L\\ \frac{\Delta p_z}{r\cos\delta}\end{bmatrix} = \frac{\mathbf{b}_L^v}{||\mathbf{b}_{L_{xy}}^v||}, \tag{2.30}$$

$$\hat{\mathbf{b}}_R^v = \begin{bmatrix}\cos\Psi_R\\ \sin\Psi_R\\ \frac{\Delta p_z}{r\cos\delta}\end{bmatrix} = \frac{\mathbf{b}_R^v}{||\mathbf{b}_{R_{xy}}^v||}. \tag{2.31}$$

In a similar way, we can define a normalized bearing to the center point of the cloud

$$\hat{\mathbf{b}}_c^v = \begin{bmatrix}\cos\Psi_c\\ \sin\Psi_c\\ \frac{\Delta p_z}{r}\end{bmatrix}. \tag{2.32}$$

Given the bearings to the outer points, we calculate the corresponding headings Ψ_L, Ψ_R, and the third component of the bearing to the center point, scaled by $\cos\delta$

$$\Psi_L = \arctan\left(\frac{\hat{b}_{L_y}}{\hat{b}_{L_x}}\right), \tag{2.33}$$

$$\Psi_R = \arctan\left(\frac{\hat{b}_{R_y}}{\hat{b}_{R_x}}\right), \tag{2.34}$$

$$\frac{\Delta p_z}{r\cos\delta} = \hat{b}_{L_z}^v = \hat{b}_{R_z}^v. \tag{2.35}$$

Following Figure 2.10, the relation between the headings Ψ_L, Ψ_R and the angle δ is defined by

$$\delta = \frac{\Psi_R - \Psi_L}{2}. \tag{2.36}$$

Since we know the angle δ, the bearing $\hat{\mathbf{b}}_c^v$ becomes function of measured quantities

$$\hat{\mathbf{b}}_c^v = \begin{bmatrix} \cos\Psi_c \\ \sin\Psi_c \\ \frac{\Delta p_z}{r} \end{bmatrix} = \begin{bmatrix} \cos\left(\frac{\Psi_L+\Psi_R}{2}\right) \\ \sin\left(\frac{\Psi_L+\Psi_R}{2}\right) \\ \cos\delta \cdot \frac{\Delta p_z}{r\cos\delta} \end{bmatrix}. \tag{2.37}$$

Given Equation (2.37), we can reconstruct the vision measurement of the center point by mapping the bearing $\hat{\mathbf{b}}_c^v$ onto the image plane

$$\begin{bmatrix} \mathbf{y}_c \\ 1 \end{bmatrix} = g\left(\mathbf{y}_{LR}\right) = \mathbf{C}\frac{\hat{\mathbf{b}}_c^{cam}}{\hat{b}_{c_z}^{cam}} = \frac{\mathbf{C}\mathbf{R}_{cam\leftarrow v}\hat{\mathbf{b}}_c^v}{\mathbf{R}_{cam\leftarrow v}\left(3,:\right)\hat{\mathbf{b}}_c^v}, \tag{2.38}$$

with $\mathbf{R}_{cam\leftarrow v}\left(3,:\right)$ specifying the third row of the matrix.

2.4. Estimator Design

The updraft position and parameter estimation includes nonlinearities in the measurement model. A variety of filters has been proposed to cope with such nonlinearities. A common property of nearly all these methods is the idea of providing a least squares estimate of the process' state. The standard approach for nonlinear estimation is the Extended Kalman Filter (EKF). As explained in Chapter 1, this filter suffers from one major flaw: If the linearized model used in the state propagation and measurement prediction steps is only a poor approximation of the true system dynamics, large errors can be introduced in the covariance transformation [53, 54, 55].

Decision-making in autonomous cross-country soaring relies on accurate updraft estimates and can be improved by additionally considering estimation uncertainties. However, in case of inconsistent state and covariance estimates, the decision-making is biased which can lead to suboptimal flight performances, or even outlandings. A common way to cope with this known flaw of the EKF is to add stabilizing noise to the state covariance after each update. This procedure is however both iterative and laborious and does not guarantee better results [53].

Particle filters, where uncertainties are transformed by propagating a "large" set of randomly

generated points through the system equations, provide a powerful alternative to the EKF and can also cope with non-Gaussian distributions. However, as reported in [56], the real-time application of these methods is questionable, especially when dealing with states of higher dimensions. The technique of Unscented Transform (UT), as introduced by the author of [57], is yet another alternative for transforming uncertainties through nonlinear functions. The UT has similarities to the particle-based transform. The selection of particles is however not conducted randomly, and only a few number of particles are necessitated to approximate Gaussian distributions. Thereby, the UT balances between the flaws of uncertainty propagation through linearization and the required computational effort of particle filters. The authors of [42] applied the UT concept to the Kalman Filter which led to the so-called Unscented Kalman Filter (UKF). As demonstrated in [58], the UKF has the capacity to provide more consistent estimates than the EKF does. Reasoning from these facts, the UKF is a well-suited candidate for the remote updraft estimation, when aiming to include information on estimated updraft uncertainties in the decision-making process.

2.4.1. Recall on Unscented Kalman Filtering

For the sake of completeness, the Unscented Kalman Filter (UKF) algorithm, as stated by the author of [59], is shortly recalled in the following.

System equations: Consider a discrete process model of type

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_k, \boldsymbol{\nu}_{\mathbf{x},k}) \qquad \text{with} \qquad \boldsymbol{\nu}_{\mathbf{x},k} \sim \mathcal{N}(0, \Sigma^2_{\mathbf{x},k}), \tag{2.39}$$

with the state vector $\mathbf{x}$, the state transition function f, the system input $\mathbf{u}$, and process noise $\boldsymbol{\nu}_{\mathbf{x}}$. The measurement model relates the state $\mathbf{x}$ to the output $\mathbf{y}$ such that

$$\mathbf{y}_k = h(\mathbf{x}_k, \mathbf{u}_k, \boldsymbol{\nu}_{\mathbf{y},k}) \qquad \text{with} \qquad \boldsymbol{\nu}_{\mathbf{y},k} \sim \mathcal{N}(0, \Sigma^2_{\mathbf{y},k}). \tag{2.40}$$

As any Kalman filter, the UKF follows the classical prediction-correction procedure.

Prediction: The UKF relies on a set $\boldsymbol{\mathcal{X}}$ of $(2L+1)$ sigma points which is calculated at each execution

$$\boldsymbol{\mathcal{X}}_{k-1} = \left[\bar{\mathbf{x}}_{k-1} \quad \bar{\mathbf{x}}_{k-1} + \sqrt{L+\lambda}\sqrt{\mathbf{P}^{\hat{\mathbf{x}}}_{k-1}} \quad \bar{\mathbf{x}}_{k-1} - \sqrt{L+\lambda}\sqrt{\mathbf{P}^{\hat{\mathbf{x}}}_{k-1}}\right], \tag{2.41}$$

where $\mathbf{P}^{\hat{\mathbf{x}}}$ is the estimated state covariance, and L indicates state dimension. Also note that each column of the matrix $\boldsymbol{\mathcal{X}}$ represents one sigma point. The constant parameter λ is function of the spread of the sigma points α and the state dimension L

$$\lambda = L(\alpha^2 - 1). \tag{2.42}$$

All sigma points are transformed through the nonlinear system function f

$$\boldsymbol{\mathcal{X}}^{(i)}_{k|k-1} = f\left(\boldsymbol{\mathcal{X}}^{(i)}_{k-1}, \mathbf{u}_k, \mathbf{0}\right) \text{ for } i = 1, ..., 2L+1, \tag{2.43}$$

where the superscript i denotes the i^{th} column of the matrix and $\boldsymbol{\mathcal{X}}_{k|k-1}$ is a matrix of predicted sigma points. The subscript $k|k-1$ in Equation (2.43) stands for a prediction at time step k, based on inputs $\mathbf{u}$, states, and measurement prior to k. The predicted average state $\bar{\mathbf{x}}_{k|k-1}$ and its corresponding covariance $\mathbf{P}^{\bar{\mathbf{x}}}_{k|k-1}$ are captured using the constant weight vectors η^c_i, η^m_i

$$\bar{\mathbf{x}}_{k|k-1} = \sum_{i=1}^{2L+1} \eta^m_i \boldsymbol{\mathcal{X}}^{(i)}_{k|k-1} \tag{2.44}$$

$$\mathbf{P}^{\bar{\mathbf{x}}}_{k|k-1} = \mathbf{Q} + \sum_{i=1}^{2L+1} \eta^c_i \left(\boldsymbol{\mathcal{X}}^{(i)}_{k|k-1} - \bar{\mathbf{x}}_{k|k-1}\right)\left(\boldsymbol{\mathcal{X}}^{(i)}_{k|k-1} - \bar{\mathbf{x}}_{k|k-1}\right)^T, \tag{2.45}$$

where $\mathbf{Q} = \Sigma^2_{\mathbf{x}}$ specifies the process noise covariance. The two weight vectors are defined as

$$\eta^c_i = \begin{cases} \frac{\lambda}{L+\lambda} + 1 - \alpha^2 + \beta & \text{, if } i = 1, \\ \frac{1}{2\,(L+\lambda)} & \text{, otherwise.} \end{cases} \tag{2.46}$$

$$\eta^m_i = \begin{cases} \frac{\lambda}{L+\lambda} & \text{, if } i = 1, \\ \eta^c_i & \text{, otherwise.} \end{cases} \tag{2.47}$$

with β being the secondary scaling constant. The sigma points $\boldsymbol{\mathcal{X}}^{(i)}_{k|k-1}$ are processed through the nonlinear measurement equation which yields the predicted observations

$$\boldsymbol{\mathcal{Y}}^{(i)}_{k|k-1} = h\left(\boldsymbol{\mathcal{X}}^{(i)}_{k|k-1}, \mathbf{u}_k, \mathbf{0}\right), \text{ for } i = 1, ..., 2L+1. \tag{2.48}$$

The mean of the predicted measurement is defined as

$$\bar{\mathbf{y}}_{k|k-1} = \sum_{i=1}^{2L+1} \eta_i^m \boldsymbol{\mathcal{Y}}_{k|k-1}^{(i)}. \tag{2.49}$$

Summing the measurement noise covariance $\mathbf{R}$ and the covariance of the transformed state, the predicted measurement covariance is

$$\mathbf{P}_{k|k-1}^{\bar{\mathbf{y}}} = \mathbf{R} + \sum_{i=1}^{2L+1} \eta_i^c \left(\boldsymbol{\mathcal{Y}}_{k|k-1}^{(i)} - \bar{\mathbf{y}}_{k|k-1}\right) \left(\boldsymbol{\mathcal{Y}}_{k|k-1}^{(i)} - \bar{\mathbf{y}}_{k|k-1}\right)^T. \tag{2.50}$$

The prediction step is completed with the computation of the cross covariance matrix

$$\mathbf{P}_{k|k-1}^{\mathbf{xy}} = \sum_{i=1}^{2L+1} \eta_i^c \left(\boldsymbol{\mathcal{X}}_{k|k-1}^{(i)} - \bar{\mathbf{x}}_{k|k-1}\right) \left(\boldsymbol{\mathcal{Y}}_{k|k-1}^{(i)} - \bar{\mathbf{y}}_{k|k-1}\right)^T. \tag{2.51}$$

Correction: The calculation of the Kalman gain and the update of both estimated state and state covariance are performed as follows

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}^{\bar{\mathbf{x}}} \left(\mathbf{P}_{k|k-1}^{\bar{\mathbf{y}}}\right)^{-1}, \tag{2.52}$$

$$\hat{\mathbf{x}}_k = \bar{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \left(\mathbf{y}_k - \bar{\mathbf{y}}_{k|k-1}\right), \tag{2.53}$$

$$\mathbf{P}_k^{\hat{\mathbf{x}}} = \mathbf{P}_{k|k-1}^{\bar{\mathbf{x}}} - \mathbf{K}_k \mathbf{P}_{k|k-1}^{\bar{\mathbf{y}}} \mathbf{K}_k^T. \tag{2.54}$$

Note that here, the caret on the state helps distinguishing between propagated and updated state estimate.

2.4.2. Estimator Structure

The functionality of the updraft map generation is to estimate the states of the clouds with a UKF, and to transform them through the updraft-cloud relation Equation (2.21) to obtain the updraft state. However, Kalman Filters require a state initialization and are prone to initialization errors.

The cloud position $\mathbf{p}_c$ can be initialized using only one vision measurement $\mathbf{y}_{LR}$, when assuming that some initial knowledge on the cloud base altitude is available. Initial guesses on the three parabola parameters are also required. The initialization of the parabola parameters necessitates information on the sign of the cloud growth, i.e whether the cloud is in a growing, or in a decaying stage. Contrary to the cloud position state, this information cannot be

deduced from one single measurement, but requires a series of consecutive measurements. To cope with these varying initialization requirements, the cloud state estimation is divided in two filters, as illustrated in Figure 2.11. The first one provides an estimate of the cloud position, and the second one estimates the cloud size parameters, i.e. the parabola. Supposing the parameters $z_i, w^\star$ and κ are available from a meteorological weather forecast, the vectors $\mathbf{d}$ can be transformed into both the vanish time and the updraft strength.
The functionality of the two estimators is described in the following.

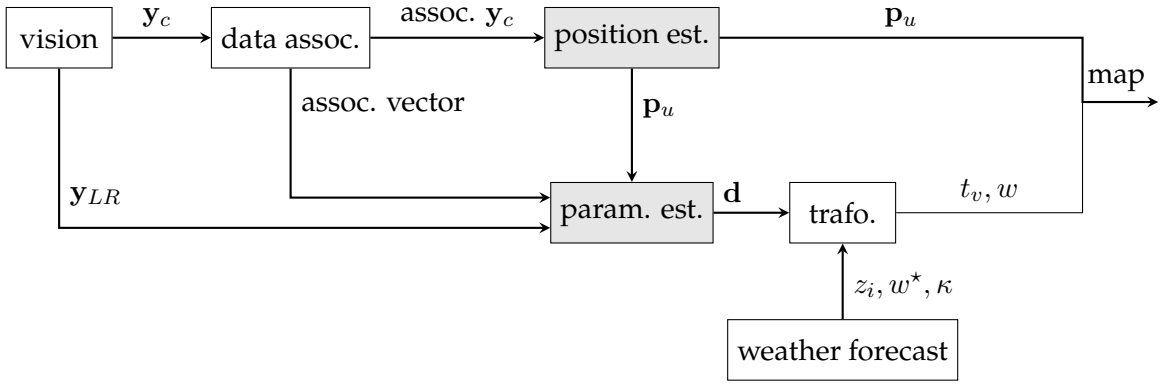


Figure 2.11: Structure of the estimator

2.4.3. Cloud Position Estimation

State Definition During thermal centering mode, the glider constantly changes its heading while it performs a spirally shaped flight trajectory. This characteristic of a thermal centering flight is shown in the flight trajectory record, illustrated in Figure 2.12.

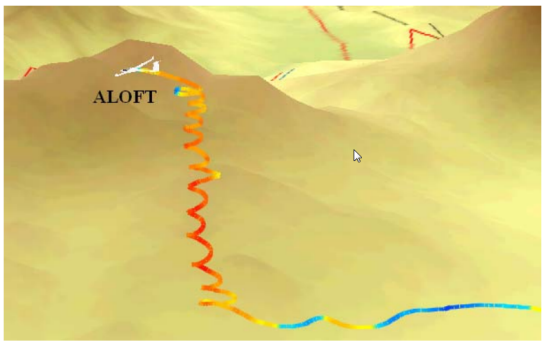


Figure 2.12: Spiral-shaped flight trajectory during thermal centering flight, [3]

As a consequence, vision measurements for individual clouds are not constantly available since the camera features a limited field of view. That is, with irregularly available vision measurements, the estimated states of single clouds cannot be updated regularly. This inherently in-

creases the filter's convergence time.

In order to compensate for these unfavorable measurement dynamics effects, induced by the glider's flight trajectory, it is important to exploit all knowledge that is available on the updrafts. Therefore, instead of running an independent filter for the estimation of each cloud position and neglecting the correlation between the individual cloud base altitudes, a combined state formulation allows a simple exploitation of the fact that all clouds lie on the same altitude level z_{cb}. It becomes thus possible to update the estimates of all cloud positions, as soon as a single vision measurement $\mathbf{y}_{LR}$ is available for one of the clouds. In other terms, the position estimate of a cloud can be updated without a vision measurement for that particular cloud. That said, the combined cloud position state estimate is defined as follows

$$\mathbf{X}_p = \begin{bmatrix} \mathbf{p}_{c_{xy,1}} \\ \mathbf{p}_{c_{xy,2}} \\ \vdots \\ \mathbf{p}_{c_{xy,n}} \\ z_{cb} \end{bmatrix}, \tag{2.55}$$

where the subscript p in the concatenated state vector $\mathbf{X}_p$ refers to the position, $\mathbf{p}_{c_{xy,i}}$ is the horizontal position of a single cloud i, and z_{cb} stands for the common cloud base altitude.

Process Model In the case of no horizontal wind, the cloud position vector remains constant between time steps. Therefore, the process model is defined as

$$\dot{\mathbf{X}}_p = f(\boldsymbol{\nu}_{\mathbf{x}\mathbf{X}_p}) = \boldsymbol{\nu}_{\mathbf{x}\mathbf{X}_\mathbf{p}}, \tag{2.56}$$

where $\boldsymbol{\nu}_{\mathbf{x}\mathbf{X}_p} \in \mathbb{R}^{2n+1}$.

Measurement Model Vision measurements $\mathbf{y}_c$ of the clouds center point are predicted from estimated cloud positions $\mathbf{p}_c = [\mathbf{p}_{c_{xy}}^T \ z_{cb}]^T$ using the camera model from Equation (2.22)

$$\mathbf{y}_c = \mathbf{I}_{2\times 3}\left(\frac{1}{\mathbf{p}_{c,z}^{cam}}\mathbf{C}\mathbf{R}_{cam\leftarrow v}\left(\mathbf{p}_c - \mathbf{p}\right)\right).$$

If the image processing algorithm detects m clouds, the true measurement vector is

$$\mathbf{Y}_c = \begin{bmatrix} \mathbf{y}_{c,1} \\ \mathbf{y}_{c,2} \\ \vdots \\ \mathbf{y}_{c,m} \end{bmatrix}. \tag{2.57}$$

Data association Data association is required to identify whether an incoming vision measurement belongs to one of the previously initialized cloud states, or originates from a newly detected cloud. A gated nearest neighbor approach, based on the Mahalabonis distance [60] is used to tackle the data association problem for vision measurements. The underlying idea is to compare the probability that a predicted measurement corresponds to an incoming measurement. This technique has proven to work reliably [61, 58], provided that the uncertainty of the predicted measurement is sufficiently small.
At each filter execution, a score s is calculated for the $n \times m$ constellations between m incoming measurements and n initialized cloud position states

$$s_{i,j} = \left[\mathbf{y}_{c,i} - \bar{\mathbf{y}}_{c,j}\right]^T \mathbf{P}^{\bar{\mathbf{y}}^{-1}_{c,j}} \left[\mathbf{y}_{c,i} - \bar{\mathbf{y}}_{c,j}\right]. \tag{2.58}$$

Scores s that feature a lower value than a threshold $s^\star$ are considered as a match. Non assigned measurements are considered to result from newly detected clouds and used to initialize a new cloud position $\mathbf{p}_{c_{xy}}$ which is appended to the combined state.

State and State Covariance Initialization Each time a new cloud is detected, both its estimated position and covariance have to be initialized. As the clouds lie on the horizontal plane which is defined by the cloud base altitude z_{cb}, it is straightforward to calculate the initial cloud position as the plane-line intersection between the cloud base plane and the line defined by the bearing $\mathbf{b}^v_c$

$$\mathbf{p}_{c_{xy}} = f(\mathbf{y}_c, z_{cb}) = \mathbf{I}_{2\times 3}\left(\mathbf{p} + \frac{z_{cb}\mathbf{p}_z}{b^v_{c_z}}\,\mathbf{b}^v_c\right). \tag{2.59}$$

For the very first cloud, this procedure requires some initial estimate of the cloud base altitude z_{cb} and its uncertainty. The initial cloud base altitude can be obtained from the weather forecast, or, if temperature and dew point measurements are available, by means of Equation (2.14). Subsequent cloud position estimates are initialized using the estimated cloud base altitude. The related state covariance matrix is obtained by applying a UT. Therefore, the 3D

measurement uncertainty of the vector $\left[\mathbf{y}_c^T \; z_{cb}\right]^T$ is transformed through Equation (2.59).

2.4.4. Cloud Size Dynamics Estimation

For reasons discussed in Section 2.4.2, the remote updraft estimation is subdivided in two filters, the first dedicated to the estimation of cloud positions, and the second for the estimation of the cloud size parameters. In the following, the second estimator is derived for the three parameters that define the cloud's parabolic diameter evolution function, as introduced in Equation (2.16). Given an estimate of these parameters, the vanish time and updraft strength belonging to the cloud can be obtained according to Equation (2.21).

Cloud Diameter Sensor The crucial measurement for the estimation of the cloud size vector $\mathbf{d}$, specified in Equation (2.16), is the vision measurement $\mathbf{y}_{LR}$. It contains the information on the current cloud size. A prediction of this four dimensional measurement vector requires the full cloud state from Equation (2.17). In other terms, the two baseline points in the image frame can only be predicted if the cloud position is known. As a consequence, the state of the cloud size estimation has to contain the cloud position. Since the cloud position is already estimated by the first filter (see Section 2.4.3), this causes an unnecessary computational load.
As an alternative, a smaller state and output vector, that can be predicted from the cloud size state $\mathbf{d}$ only, is obtained by introducing a virtual cloud diameter sensor

$$d = 2r \sin \delta, \tag{2.60}$$

where

$$r = ||\mathbf{p}_{c_{xy}} - \mathbf{p}_{xy}||. \tag{2.61}$$

Note that r and δ from Equation (2.60) are known. More specifically, the horizontal distance to the cloud can be determined from the estimated cloud position which is provided by the first filter, and the angle δ is obtained from the vision measurement $\mathbf{y}_{LR}$, (see Equations (2.33), (2.34) and (2.36)). Therefore, the measurement for the cloud parameter estimator is function of the state of the position estimation and the vision measurement, i.e. $y_d = f\left(\mathbf{y}_{LR}, \mathbf{p}_c\right)$.
Equation (2.60) thus specifies a virtual sensor that depends on known quantities. Its level of output uncertainty σ_{y_d} is defined by the uncertainty of the feeding signals which are the covariance of the cloud position estimate $\mathbf{P}^{\mathbf{p}_c}$ and the vision measurement noise $\Sigma_{\mathbf{y}_{LR}}$. The technique of UT, as discussed in Section 2.4.1, is used to obtain the measurement noise covariances of the

inferred cloud diameters by transforming the input uncertainties through the corresponding sensor equations.

State Definition While there exists a correlation between the cloud positions, the size parameters of different clouds are uncorrelated. That said, the cloud size parameter estimation is implemented by running n filters in parallel, where the state of a single cloud is defined in Equation (2.16)

$$\mathbf{d} = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix}.$$

Process Model Since the state $\mathbf{d}$ is assumed to be time invariant, the process model is a function of the process noise $\boldsymbol{\nu}_{\mathbf{x_d}}$ only

$$\dot{\mathbf{d}} = \boldsymbol{\nu}_{\mathbf{x}_d}. \tag{2.62}$$

Measurement Model Cloud diameter measurements are related to the state according to Equation (2.15)

$$y_d = e_1 t^2 + e_2 t + e_3.$$

State and State Covariance Initialization Kalman Filters are sensitive to initialization parameters in the way that the initial covariance and estimation error have to be consistent. In order to initialize the three parameters of the parabola, at least three constraints about the parabola have to be known a priori. A rigorous initialization of the state requires some guess on the slope of the diameter at the time the filter is initialized. This information cannot be obtained from only one diameter measurement. Such initialization problems also occur in other domains, for example radar-based target tracking, where the velocity of the target object cannot be initialized given only one distance-measurement [62].

Different methods for obtaining the slope of the diameter can be thought of. One approach is to batch-fit a series of collected cloud measurements, as suggested in [62]. This solution however requires an increased memory consumption. As an alternative, a recursive initialization period is proposed, where the slope of the cloud diameter is estimated by a linear Kalman Filter. The state of the linear filter is defined as $\mathbf{x}_s$ and contains the slope μ and the cloud diameter

at the beginning of the estimation d_0

$$\mathbf{x}_s = \begin{bmatrix} \mu \\ d_0 \end{bmatrix}, \tag{2.63}$$

Cloud diameter measurements are predicted according to

$$y_d = d_0 + \mu(t - t_0), \tag{2.64}$$

where t_0 is the time at the beginning of the slope estimation.
A UKF with the state Equation (2.62) is triggered on the condition that the slope μ is positive, and its covariance inferior to a threshold value $\gamma^\star$. Given the estimated slope μ, the current diameter d, and time t as well as some guess about the maximum diameter of the cloud d_{max}, the three parameters of the parabola are initialized according to

$$e_1 = -\frac{\mu}{4(d_{max} - d)}, \tag{2.65}$$

$$e_2 = \mu - 2e_1 t, \tag{2.66}$$

$$e_3 = d - e_1 t^2 - e_2 t. \tag{2.67}$$

The initial state covariance $\mathbf{P}^{\mathbf{x_d}}$ is obtained by applying an UT of the parameters d, μ and d_{max} and their corresponding covariances through Equations (2.65) to (2.67).

2.4.5. Data Flow

The flowchart illustrated in Figure 2.13 gives an overview of the estimator's functionality. It points out the parallel structure of the filter, separating the position estimation from the cloud size parameter estimation.
The entry point is the vision block. It provides the pixel positions of the two cloud feature points $\mathbf{Y}_{LR}$ for all identified clouds which are transformed into vision measurements $\mathbf{Y}_c$. The latter are directed to the data association which calculates whether a measurement originates from an already initialized (in.), or from an uninitialized cloud (unin.). The correction/update step is conducted with measurements from initialized clouds. Uninitialized measurements $\mathbf{Y}_{c,k}^{\text{unin.}}$ are used to initialize new cloud positions. The estimated state is augmented by these positions. Therefore, the dimension of $\mathbf{X}_p$ is time-varying. At the next filter call, state predictions are conducted with the augmented state.

The vector $\boldsymbol{\Theta}$ contains data association information, necessary to separate between cloud width measurements $\mathbf{Y}_d^{\text{in.}}$ and $\mathbf{Y}_d^{\text{unin.}}$. The vector $\mathbf{X}_d = [\mathbf{d}_1, \dots \mathbf{d}_l]$ specifies a filter bank, i.e. a set of parameter filters that run in parallel. For newly initialized states, the filter bank is extended. Cloud positions $\mathbf{X}_p$ and cloud parameter states $\mathbf{X}_d$ are transmitted to the updraft cloud relation block where $\mathbf{X}_u$ is generated. Note that updrafts for which the parameter estimation has not yet been triggered are not considered in the updraft map. Therefore, the map contains l updrafts. This state reduction is not illustrated in the flowchart for the sake of easier readability.

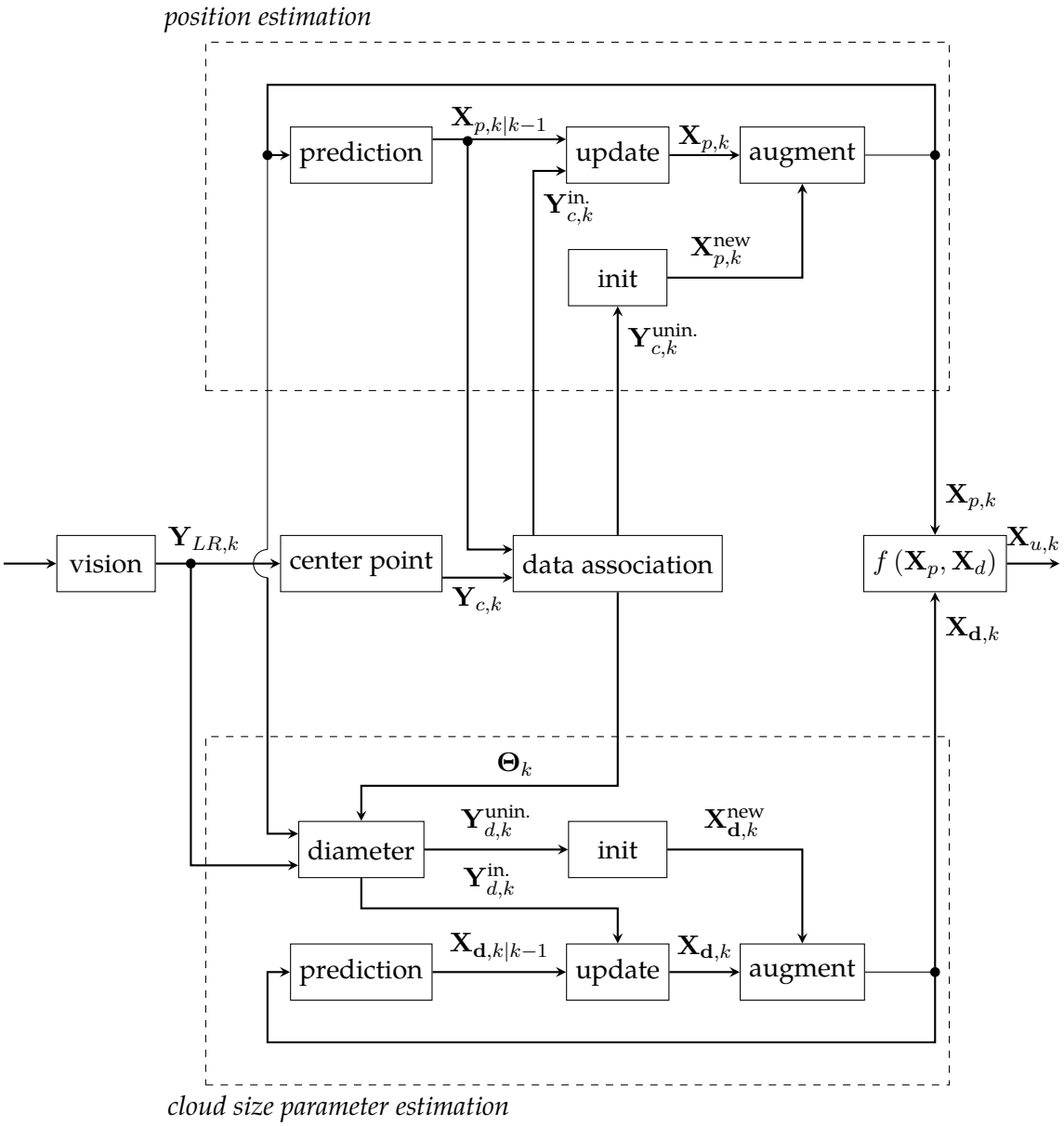


Figure 2.13: Data flow of the estimation process

2.5. Simulation Results

This section presents computer simulation results. The main purpose of the simulations is to study the filter performance in terms of estimation consistency and convergence.

2.5.1. Context

Kalman Filters not only provide an estimate of the state, but also the estimated state covariance. The state covariance constitutes a metric of the accuracy of the estimation. As will be discussed in Chapter 3, it is beneficial to include this supplemental information in the path planning task, to render the decision-making process more robust. However, this requires consistent state covariance estimates. An inconsistent ("underconfident" or "overconfident") estimate can result in less efficient or high-risk trajectories. Applied to path planning for autonomous cross-country soaring, an inconsistent estimation would result in lower cross-country speeds, or, even worse, in an increased number of outlandings. Therefore, it is important to not only study the convergence of the filter, but also its consistency.

2.5.2. Parameter Values

The parameters required for the estimator are the ones of the camera, and the filter/sensor settings.
The camera characteristics that were defined in Section 2.3 are specified in Table 2.1. They represent a typical low-cost RGB camera.

Parameter	Value	Unit
x_0	980	[px]
y_0	640	[px]
S_x	454550	[px/m]
S_y	454550	[px/m]
f	0.76871	[mm]

Table 2.1: Camera characteristics

The filter parameters include the process and measurement noise covariances, as well as the

sampling time. Both the process and measurement noise covariance are defined, such that

$$\begin{aligned} \mathbf{Q} &= \mathbb{E}\left[\boldsymbol{\nu}_{\mathbf{x}}\boldsymbol{\nu}_{\mathbf{x}}^T\right] &= \Sigma_{\mathbf{x}}^2, \\ \mathbf{R} &= \mathbb{E}\left[\boldsymbol{\nu}_{\mathbf{y}}\boldsymbol{\nu}_{\mathbf{y}}^T\right] &= \Sigma_{\mathbf{y}}^2. \end{aligned} \tag{2.68}$$

Recall from Equation (2.25), that the diagonal matrices Σ specify the standard deviations. The corresponding parameter values are listed in Tables 2.2 and 2.3.

Parameter	Value	Unit
$\Sigma_{\mathbf{x}_{\mathbf{p}_{c_{xy}}}}$	$\mathbf{0}_{2\times2}$	[m/s]
$\Sigma_{\mathbf{x}_{z_{cb}}}$	0	[m/s]
$\Sigma_{\mathbf{x_s}}$	$\mathbf{I}_{2\times2}\begin{bmatrix}0.005 & 0.1\end{bmatrix}^T$	[m/s^2], [m/s]
$\Sigma_{\mathbf{x_d}}$	$\mathbf{I}_{3\times3}\begin{bmatrix}0 & 0 & 0.02\end{bmatrix}^T$	[m/s^3], [m/s^2], [m/s]

Table 2.2: Process noise parameter values

Parameter	Value	Unit
$\Sigma_{\mathbf{y}_{LR}}$	$2\mathbf{I}_{4\times4}$	[px]

Table 2.3: Measurement noise parameter values

Note that the units refer to the corresponding diagonal elements of the Σ-matrices. In case the same unit applies to several entries, the unit is only noted once. As mentioned earlier, model uncertainties are not considered in this work. Thus, most of the process noise covariance entries are zero. Furthermore, note that the performance of the image processing algorithms, that were suggested in [33], have not yet been assessed. That said, in this work, the standard deviation of the vision measurement $\mathbf{y}_{LR}$ is assumed to be $2\,$px. An overall system evaluation that also considers increased measurement noise is conducted in Chapter 4.

The remaining filter parameters are noted in Table 2.4. It specifies the data association threshold for matches $s^\star$ (see Section 2.4.3), and the threshold covariance $\gamma^\star$ (see Section 2.4.4) used to trigger the parabola tracking.

The sample time of the two filters is selected to be $0.1\,$s.

Parameter	Value	Unit
$s^\star$	5	[-]
$\gamma^\star$	0.085^2	[m/s]^2

Table 2.4: Threshold parameter values

2.5.3. Evaluation Metrics

A criterion that is dedicated to study the consistency of an estimator is the so-called Normalised Estimation Error Squared (NEES), noted ϵ. Following the definition from the author of [63], the NEES at sample k is defined as

$$\epsilon_k = \tilde{\mathbf{x}}_k^T \mathbf{P}_k^{-1} \tilde{\mathbf{x}}_k, \tag{2.69}$$

where $\tilde{\mathbf{x}}$ specifies the error between the true and the estimated state, i.e. $\mathbf{x} - \hat{\mathbf{x}}$. Under linear and Gaussian assumptions, the average of the squared norm of the estimation error satisfies the following hypothesis

$$\mathbb{E}\left[\tilde{\mathbf{x}}_k^T \mathbf{P}_k^{-1} \tilde{\mathbf{x}}_k\right] = L, \tag{2.70}$$

where L is defined as the state dimension. Equation (2.70) is true in case the estimator correctly estimates the state covariance $\mathbf{P}$.

The average NEES can be calculated from MC simulations. For a total of N simulation runs, the N-run average NEES is defined as

$$\bar{\epsilon}_k = \frac{1}{N} \sum_{i=1}^{N} \epsilon_k^i, \tag{2.71}$$

where i indicates the run.

The calculated NEES can be verified by applying a χ^2-test. The test is passed, if the NEES remains within an acceptance interval

$$\bar{\epsilon}_k \in [r_1, r_2], \tag{2.72}$$

where r_1 and r_2 specify the bounds of the acceptance interval. These bounds depend on the selected significance level α such that

$$1 - \frac{\alpha}{2} = F_{\chi^2(N \times L)}(r_1) \Rightarrow r_1 = F^{-1}_{\chi^2(N \times L)}\left(1 - \frac{\alpha}{2}\right) \tag{2.73}$$

$$\frac{\alpha}{2} = F_{\chi^2(N \times L)}(r_2) \Rightarrow r_2 = F^{-1}_{\chi^2(N \times L)}\left(\frac{\alpha}{2}\right), \tag{2.74}$$

where F stands for the Cumulative Distribution Function (CDF) which is defined as

$$F_{\chi^2(N \times L)}(x) = \int_0^x f_{\chi^2(N \times L)}(y)\, dy. \tag{2.75}$$

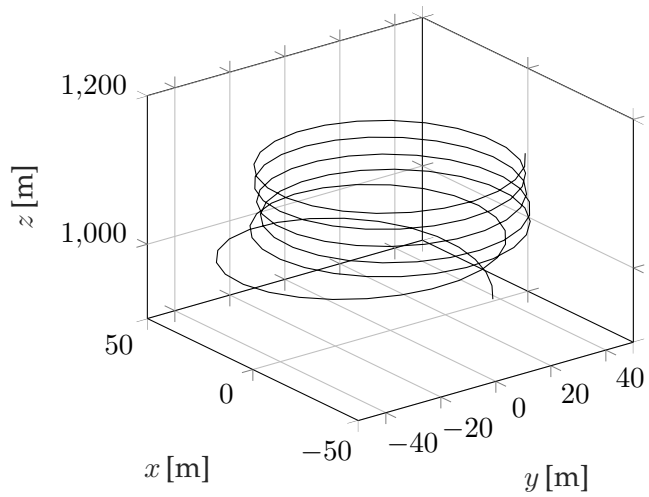
While the NEES is suitable to study the consistency, the Root Mean Square Error (RMSE) provides a metric to evaluate the estimation performance in terms of convergence. The RMSE at sample k over N simulation runs is defined such that

$$\xi_k = \sqrt{\frac{1}{N}\sum_{i=1}^{N} \tilde{\mathbf{x}}_k^T \tilde{\mathbf{x}}_k}. \tag{2.76}$$

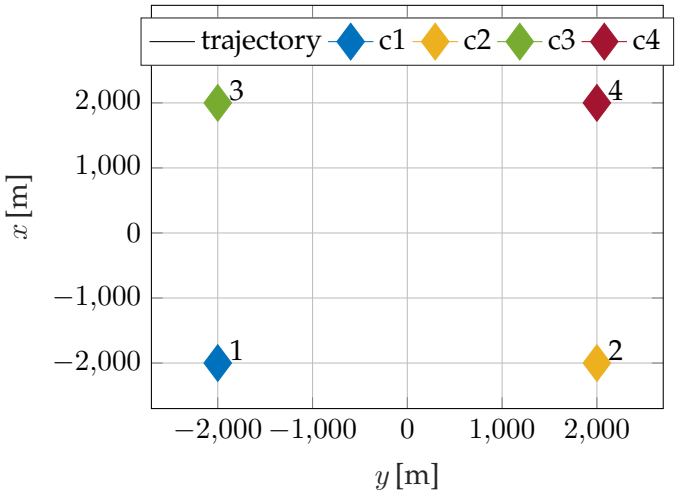
2.5.4. Monte Carlo Simulation Scenario

For obtaining representative estimation conditions, the MC scenario is specified by a typical thermal centering flight trajectory, as illustrated in Figure 2.14a. The vehicle centers an updraft at 950 m altitude and climbs to the cloud base at about 1200 m altitude. The flight trajectory is generated relying on the aircraft model and the flight control laws that are described in Appendices A and B. The vehicle circles around the origin of the xy frame, where the lowest circle in Figure 2.14a is slightly shifted with respect to the upper circles. This is because the vehicle is not yet centered around the updraft's core at the beginning of the simulation.
During the climb, the updrafts of four clouds are estimated, as shown in Figure 2.14b. The rhombi in Figure 2.14 indicate the mean value of the true cloud positions. However, for each simulation run, the true cloud positions are varied, where the variation is specified by the mean and a given standard deviation.
Due to the spiral-shaped flight trajectory, clouds repeatedly appear and disappear on the image sensor, i.e. vision measurements are not constantly available. This creates realistic observation conditions in terms of measurement regularity.
It is further worth mentioning, that the observation trajectory stays representative, beyond the fact that horizontal wind is not considered. This is because horizontal wind does not change the relative motion between the glider and the observed clouds, since both updraft and clouds drift downwind with the same speed. Thereby, the relative motion is not impacted.
For the Monte Carlo simulation, initial states are imposed a priori. This allows to generate samples of the true states, according to both the initial state and state covariance, i.e. $\mathbf{x}_0$ and $\mathbf{P}_0$ respectively. At each run, the true state is sampled from this normal distribution which generates N different scenarios with

$$\begin{aligned} \mathbf{x} &\sim \mathcal{N}\left(\mathbf{x}_0, \Sigma^2_{\mathbf{x}_0}\right) \\ \mathbf{P}_0 &= \mathbb{E}\left[\mathbf{x}_0 \mathbf{x}_0^T\right] = \Sigma^2_{\mathbf{x}_0}. \end{aligned} \tag{2.77}$$



(a) Reference scenario - Aircraft/observer trajectory



(b) Reference scenario - Aircraft cloud setup

Figure 2.14: Reference scenario

The proposed estimator separates the position estimation from the updraft velocity and vanish time estimation. The latter is triggered once the slope of the cloud diameter has been detected to be positive, and its estimated covariance undercuts the threshold value $\gamma^\star$ (see Section 2.4.4). This threshold is usually reached when the cloud position estimation has converged. In other terms, the vanish time and updraft strength estimation requires the cloud position state to be converged. To account for this, two dedicated Monte Carlo simulations are conducted.

In the first one, the cloud position estimation is evaluated. Table 2.5 lists the initialization parameters for the first Monte Carlo run.

The uncertainty in the initial cloud positions (defined by the values $\Sigma_{\mathbf{p}_{c_{xy,i}}}$ and $\Sigma_{z_{cb}}$) is selected to be 150 m and 10 m respectively. This comparably large uncertainty represents the phase where the vanish time and updraft velocity estimation is not yet triggered. The mean parabola

Parameter	Value	Unit
$\mathbf{X}_p$	$1000\,\big[\underbrace{2 \quad 2}_{\mathbf{p}_{c_{xy,1}}}\ \underbrace{2 \quad -2}_{\mathbf{p}_{c_{xy,2}}}\ \underbrace{-2 \quad -2}_{\mathbf{p}_{c_{xy,3}}}\ \underbrace{-2 \quad 2}_{\mathbf{p}_{c_{xy,4}}}\ \underbrace{-1.2}_{\mathbf{z}_{cb}}\ \big]^T$	[m]
$\Sigma_{\mathbf{p}_{c_{xy,i}}}$	$150\,\mathbf{I}_{2\times 2}$	[m]
$\Sigma_{z_{cb}}$	10	[m]
$\mathbf{d}_0$	$\begin{bmatrix}-0.0005 & 0.8333 & 416.667\end{bmatrix}^T$	[m/s^2],[m/s],[m]
$\Sigma_{\mathbf{d}}$	$0.1\mathbf{d}_0\mathbf{I}_{3\times 3}$	[m/s^2], [m/s], [m]

Table 2.5: Initialization parameters for the first Monte Carlo run

coefficients e_1, e_2 and e_3 contained in the vector $\mathbf{d}$ are selected to generate clouds with a lifespan of 40 min and a maximum size d_{max} of 750 m on average. Their standard deviation is set to be 10% of the mean initial value. Thus, for each cloud i at run j $\mathbf{d}_{ij} \sim \mathcal{N}\left(\mathbf{d}_0, \Sigma^2_{\mathbf{d}_0}\right)$. The vertical velocity scale $w^\star$ is set to 3 m/s and κ is $1/\bar{d}_{max}\bar{w}_{u,rz} = 0.0033\,1/\text{s}$ where $\bar{w}_{u,rz}$ is 2.455 m/s for a minimum altitude of 500 m and a cloud base altitude of 1200 m.

For the second MC simulation run, the initial uncertainties in the cloud position states are selected from a normal distribution with less uncertainty, as noted in Table 2.6. This simulates that the position estimate has converged and the vanish time and strength estimation has been triggered. The remaining settings are the same as in the first run.

Parameter	Value	Unit
$\Sigma_{\mathbf{p}_{c_{xy,i}}}$	$5\,\mathbf{I}_{2\times 2}$	[m]
$\Sigma_{z_{cb}}$	5	[m]

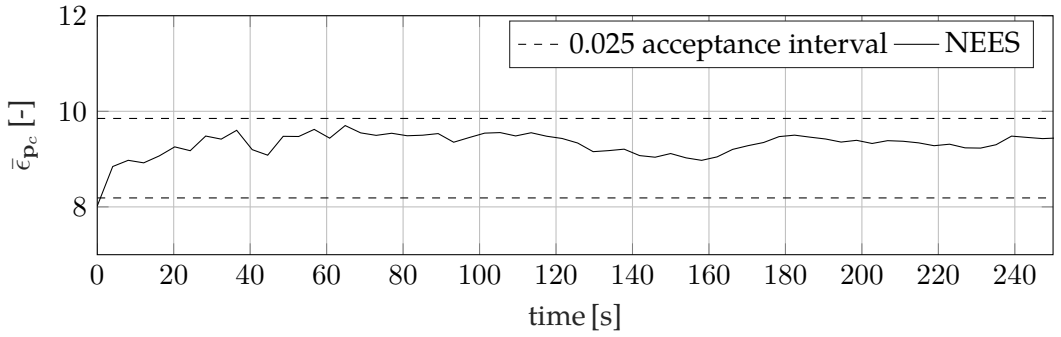
Table 2.6: Initialization parameters for the second MC run

2.5.5. Consistency and Convergence

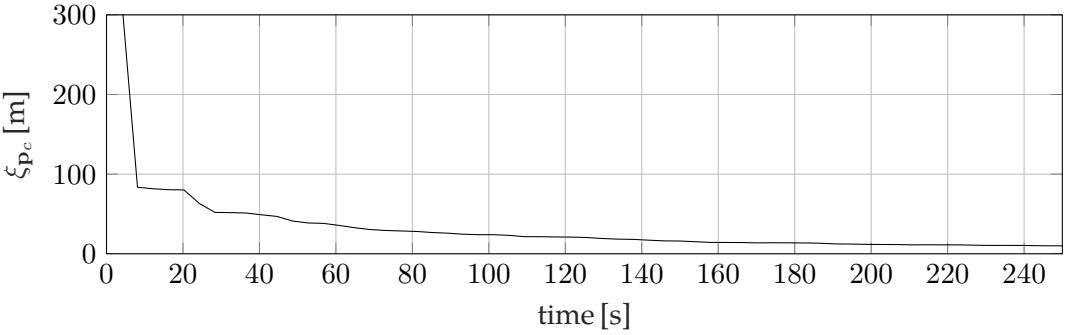
Figure 2.15 illustrates Monte Carlo results in terms of NEES and RMSE from 100 runs for the position estimation evaluation. The dashed lines in the upper graph (Figure 2.15a) represent the two sided acceptance interval for $N = 100$ runs and a state dimension of $L = 9$ for a probability concentration region of 95%. Note that the state dimension L is determined from $4 \times 2\,\text{D}$ cloud positions and the cloud base altitude z_{cb}.

The NEES remains within the acceptance bounds, i.e. the covariance of the position state is correctly estimated, despite the adverse observation conditions. The consistent position estimate provides an important result for the vanish time and updraft velicity estimation. This is because the measurement noise of the cloud diameter sensor from Equation (2.60) depends

on the uncertainty of the cloud position. Inconsistent cloud position estimates would therefore result in inconsistent estimated vanish times and updraft strengths, yielding biased path planning decisions.



(a) Average NEES



(b) RMSE

Figure 2.15: NEES and RMSE for the cloud position tracking

The lower graph illustrates the convergence characteristics of the position estimation. The initial RMSE is reduced to 10% after approximately 100 s which roughly corresponds to three circles in the thermal climb trajectory (see Figure 2.14a).

Figures 2.16 and 2.17 show the results of the second Monte Carlo simulation - dedicated to the evaluation of the updraft strength and vanish time estimation.

The four solid lines in each of the graphs specify the results of the individual clouds. Note that the graphs illustrate the NEES and RMS of the vanish time and updraft velocity, obtained after propagating the estimated parabola parameters through the updraft cloud relation.

Figure 2.16 illustrates the result in terms of consistency with the dashed lines defining the region of acceptance. For all clouds, the NEES mostly remains within the acceptance bounds, indicating that both the uncertainty of the updraft strength and vanish time are correctly estimated.

Another interesting fact is that the initial RMSE, illustrated in Figure 2.17b, decreases slowly, compared to the RMSE of the position estimation from Figure 2.15b. This is due to the fact that

the width measurements are particularly noisy.

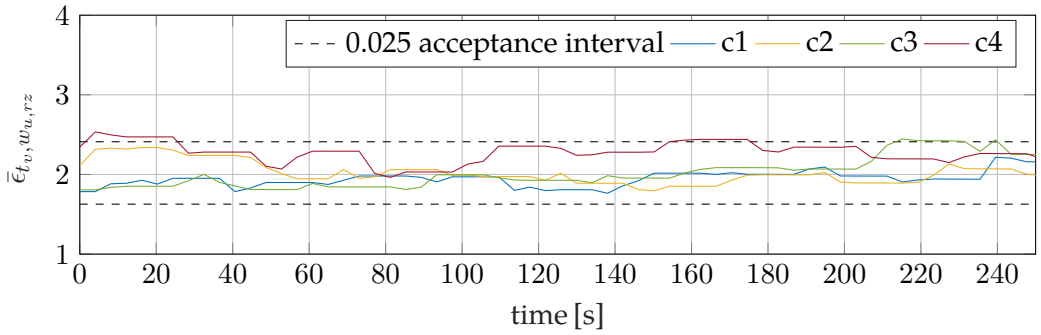
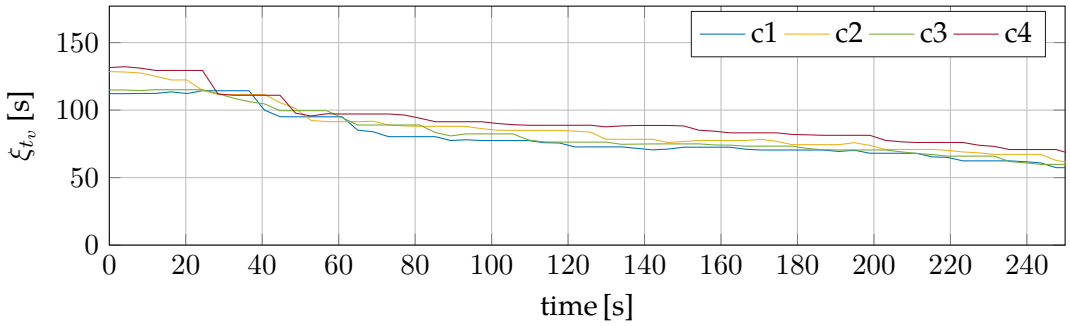
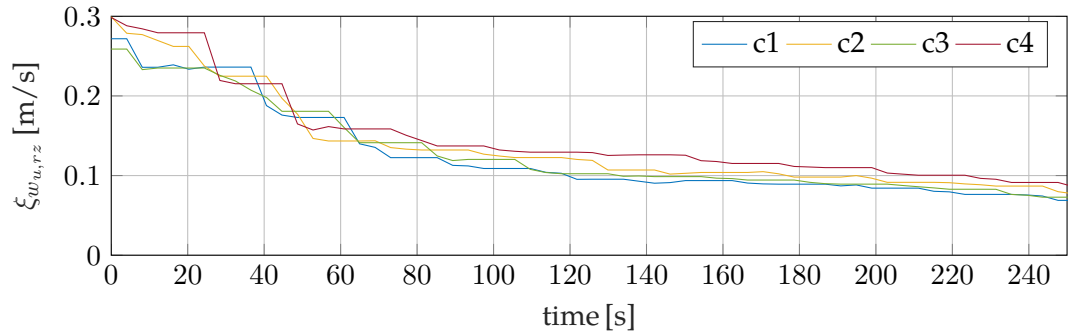


Figure 2.16: NEES for updraft vanish time and updraft strength



(a) RMSE - Estimated vanish time



(b) RMSE - Estimated climb updraft strength

Figure 2.17: RMSE for updraft vanish time and updraft strength

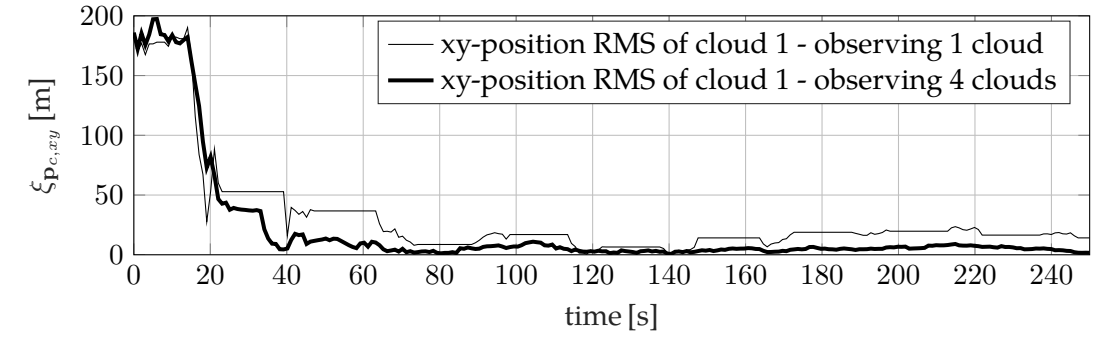
2.5.6. Benefit of the Observation of Multiple Clouds

As stated in Equation (2.14), cumulus clouds share the same cloud base altitude when presuming the temperature difference between the air and the dewpoint to remain constant over a region. This correlation in the cloud positions was exploited in the estimator design, by combining all horizontal cloud positions in one state with one common cloud base altitude, rather than estimating the 3 D cloud positions individually. The main benefit is that the position estimate of all clouds can be updated whenever vision measurements for any of the clouds are available.

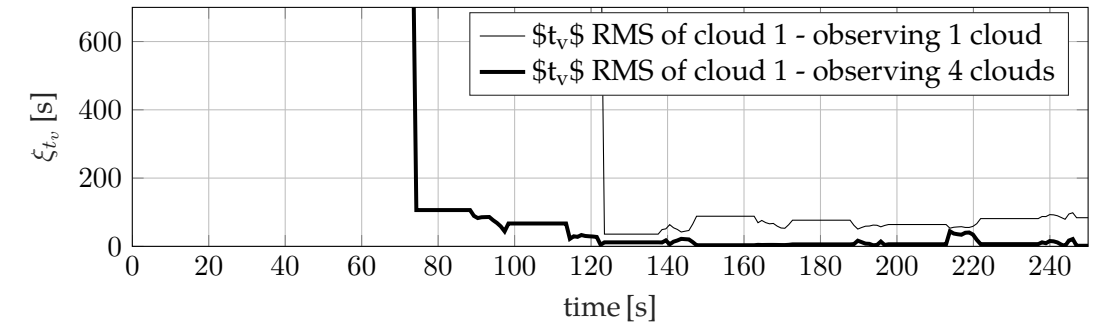
To illustrate this effect, the simulation scenario from Figure 2.14 is considered and repeated twice. In the first run, only the cloud number one is estimated, i.e. estimation updates depend on measurements of only one cloud. In the second run, all four clouds are tracked.

Figure 2.18a illustrates the RMSE of the horizontal position $\mathbf{p}_{c_{xy}}$ of cloud #1. The updraft strength and vanish time estimates of that cloud are shown in Figures 2.18b and 2.18c. Note that in both graphs, the bold line represents the case where four clouds are tracked.

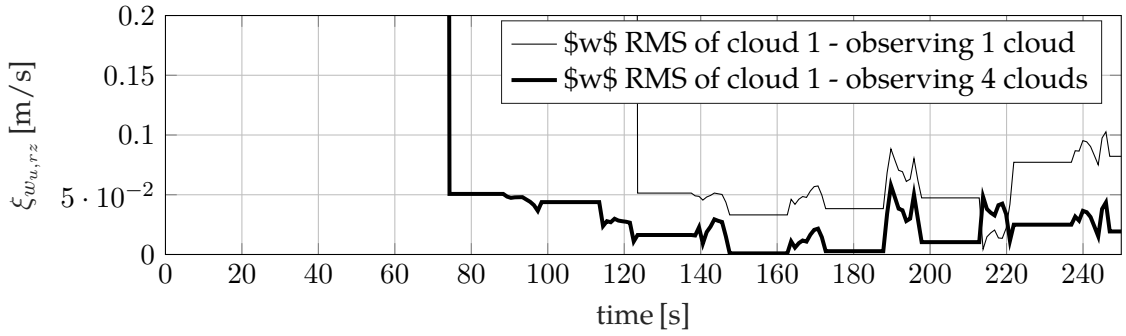
As expected, the error of the horizontal position decreases faster as the position estimate of the cloud is updated even while no vision measurements corresponding to this cloud are available. This can be seen at t =25 s where the cloud disappears for the first time from the vision sensor. For the case where only one cloud is tracked, the horizontal position error remains constant until the glider has performed a full circle and the cloud reappears in the camera's field of view. This is in contrast to the case when observing four clouds. Even while the cloud number one is not in the field of view, its horizontal position error decreases. These varying estimation performances in terms of convergence time propagate into the time at which the vanish time and updraft strength is triggered. Therefore, consider Figures 2.18b and 2.18c. In the multi-cloud run, the updraft parameter estimation of cloud 1 is triggered at t =75 s which is approximately 50 s earlier than in the single-cloud case. In other terms, information on the updraft strength and vanish time is available earlier, and the path planning can compute trajectories based on more complete information of the vehicle's environment.



(a) Estimated horizontal position



(b) Estimated vanish time



(c) Estimated updraft strength

Figure 2.18: Estimation error depending on the number of tracked clouds

2.6. Summary

This chapter describes the design of a parallel Unscented Kalman Filter (UKF) for the vision-based estimation of sub-cumulus updraft parameters.
First, a simple atmospheric model is derived, enabling the design of the vision-based updraft estimator. The estimation of updraft positions is separated from the vanish time and strength estimation for tackling the problem of varying initialization dynamics. Therefore, two parallel filters are designed. The primary focus is to generate consistent state estimates by conducting careful state initialization and uncertainty propagation using the technique of Unscented Transform (UT). The second focus is the selection of the combined cloud position state, that incorporates the fact that cumulus clouds appear on the same altitude level, thus allowing for faster filter convergence. Monte Carlo (MC) simulations are conducted to assess the performance of the estimation algorithm with respect to convergence and consistency. The simulation results reveal the capability of the proposed estimator to provide consistent and converging state estimates. This achievement allows to include estimation uncertainties in the path planning process without biasing the calculated paths due to inconsistent state estimates. Further simulations demonstrate the benefit of the combined cloud position state.

Chapter 3

Path Planning For Cross-Country Soaring

As stated in Chapter 1, path planning is a crucial task for improving the reliability of autonomous cross-country soaring vehicles. However, there has been relatively little research on that field. This is for two reasons. First, autonomous soaring is a young field of research. Second, to this date, remote updraft estimation functionality does not exist.
The authors of [47] use the Floyd-Warshall algorithm [64] to calculate an exact solution of the planning problem. In [48], a heuristic based sub-optimal solution of the planning-problem is suggested for long distance flights through areas with high updraft densities. Interestingly enough, in both contributions, full and certain a priori knowledge of all future updrafts over the entire mission horizon are supposed to be available. However, in Vision-Based Autonomous Cross-Country Soaring (VBACCS), the vehicle perceives its environment during flight and the updraft estimation depends on both the vehicle's flight trajectory and the randomness of the nature. Moreover, the remote updraft estimation as introduced in Chapter 2 provides updraft maps of limited size, determined by the estimation performance and the detection range of the image processing device. The mission destination thus may lie far beyond the horizon of the provided map and only a fragment of total flight can be calculated. Most importantly, the updraft maps employed in VBACCS are imperfect, i.e. they feature uncertainties. Ignoring these uncertainties in the planning task can increase the number of mission failures and thus restrict the applicability of VBACCS. These are the main evidences that underline the need of more advanced planning algorithms for VBACCS.
That said, in this chapter, a probabilistic path planner for VBACCS is proposed. Beyond in-

corporating uncertainties, it also provides a local planning solution in case the destination lies outside the range of the estimated map. Finally, it is shown that the suggested path planner is implementable on a low-cost, miniaturized computer platform, thus being a solution that can be embedded on-board a small glider UAV.

This chapter is organized as follows:

In Section 3.1, the map-based cross-country soaring problem is stated. Section 3.2 serves to explain the glider motion model, necessary for the design of the path planning algorithm and suggests a guidance law for the final glide. In Section 3.3, the impact of uncertain updraft estimates in cross-country soaring is explained. Section 3.4 provides an overview of planning in uncertain environments and recalls the fundamentals of Markov Decision Processes (MDPs) and dynamic programming. In Section 3.5, the cross-country soaring problem is formalized as a MDP and it is shown how its solution can be calculated. Section 3.6 presents computer simulation results that compare the proposed method to former approaches and a study of the runtime. The content of this chapter is summarized in Section 3.7.

3.1. The Path Planning Problem

As illustrated in Figure 3.1, the cross-country soaring mission for the glider consists of flying to a destination $\mathbf{p}_D$ in minimum time, using updrafts that arise during the flight. The destination is given by $\mathbf{p}_D \in \mathbb{R}^3$ where $[\mathbf{p}_{D_x}\ \mathbf{p}_{D_y}\ \mathbf{p}_{D_z}]^T$. The altitude of the vehicle at the destination $\mathbf{p}_{D_z}$ is a free parameter such that $\mathbf{p}_{D_z} \geq z_{min}$ where z_{min} is some minimum altitude. Therefore, the path planning can be seen as a constrained shortest path problem where the vehicle's vertical operational range is restricted to $z_{min} \leq \mathbf{p}_z \leq z_{cb}$. Recall from Figures 2.2 and 3.1 that z_{cb} is the cloud base altitude. Undercutting the minimum altitude z_{min} is considered as mission failure. This minimum altitude could for instance be defined by airspace restrictions.

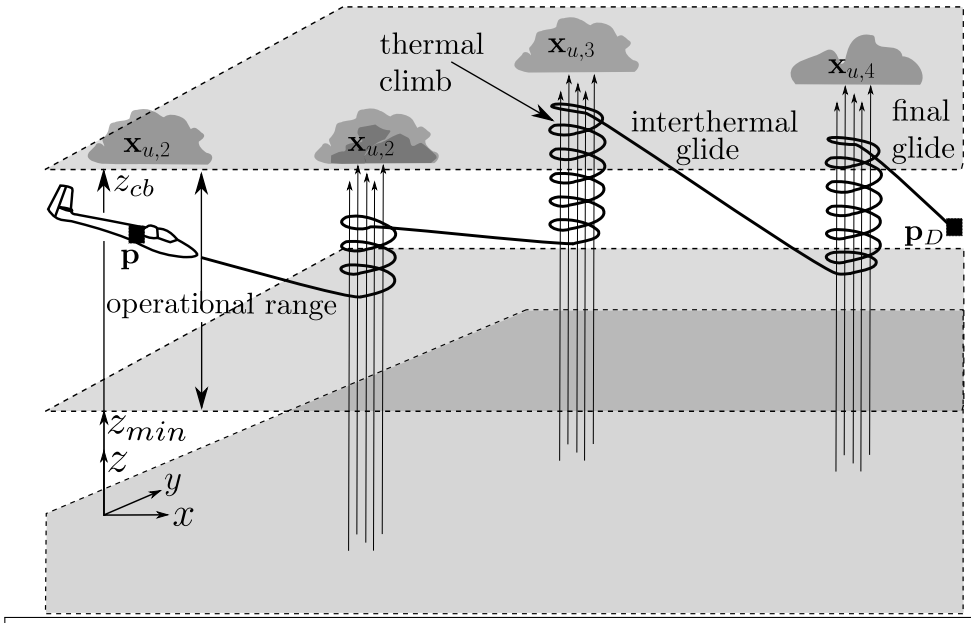
The success and performance of a cross-country soaring flight mostly depend on three factors.

1. The vehicle's aerodynamic performance. High performance sailplanes with a high glide ratio can transform altitude into traveled distance more efficiently, i.e. faster and at a lower sink rate.

2. The meteorological conditions. Both the density of updrafts and updraft velocities directly impact the flight time.

3. The decision-making of the pilot/autopilot w.r.t both the flight path and airspeed. If the vehicle is not equipped with any internal propelling system, inefficient or wrong decisions not only increase the flight time, but can lead to so-called outlandings where the vehicle has to abort the flight by landing on a site which is not the targeted one.

In this thesis, we seek to improve the cross-country soaring performance by advanced decision-making. Therefore, we have to process information on the vehicle performance and the updraft map in order to calculate a sequence of consecutive control inputs/actions that bring the vehicle to the target in minimum time, if possible, or to decide to abort the mission and land the vehicle safely. To fulfill this task, models that describe the vehicle dynamics, and its interaction with meteorological phenomena are required.



$\mathbf{p}$	glider position	$\mathbf{p}_D$	destination position	$\mathbf{x}_u$	updraft state
z_{min}	minimum altitude	z_{cb}	cloud base altitude		

Figure 3.1: Path planning problem for map-based cross-country soaring

3.2. Airspeed Selection and Glider Motion Model

For the path planning task, a motion model is required in order to propagate the glider's state. We rely on a simplified point-mass model, for the sake of computational efficiency. As illustrated in Figure 3.1, the glider can operate in three different modes which are interthermal glide, thermal climb and final glide. In cross-country soaring, the high-level control inputs (actions) are the sequence of updrafts to visit, and the corresponding airspeed commands. For

keeping the path planning problem tractable, we separate the airspeed selection from the path planning task. That is, the airspeed is not considered as a control variable in the path planning, but calculated as function of a separate guidance law. That said, this section also serves to identify suitable airspeed commands.

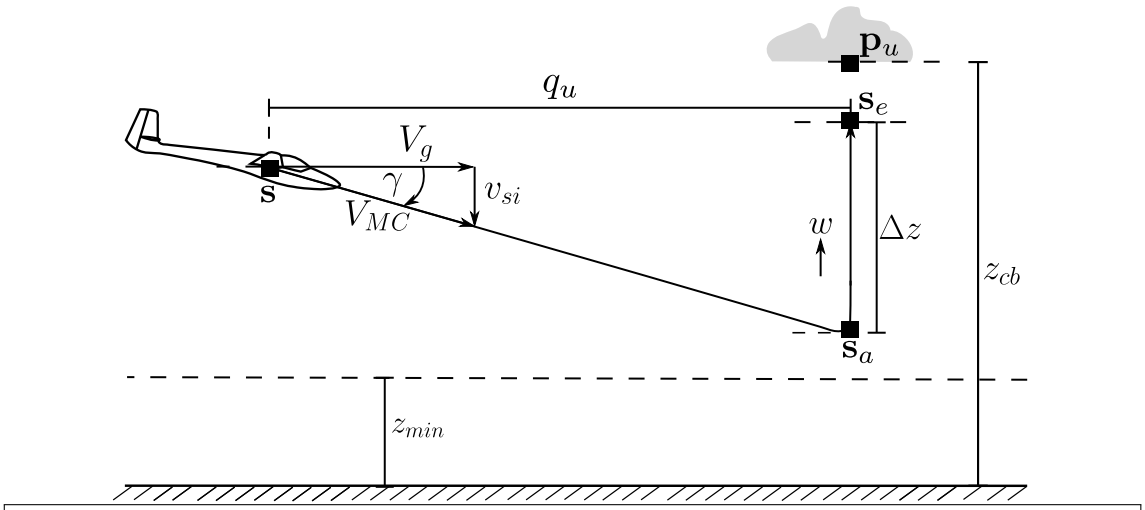
In the following, the vehicle's state $\mathbf{s}$ is defined as

$$\mathbf{s} = \begin{bmatrix} \mathbf{p} \\ t \end{bmatrix}, \tag{3.1}$$

where $\mathbf{p} = [\mathbf{p}_x\, \mathbf{p}_y\, \mathbf{p}_z]^T$ is the glider's position and t stands for the time.

3.2.1. Interthermal Glide

The maneuver for harvesting an updraft's energy can be separated in two consecutive sequences. As depicted in Figure 3.1, these sequences are interthermal glide and thermal climb. The interthermal glide is the straight and steady flight phase between some state $\mathbf{s}$ and the arrival state $\mathbf{s}_a$ underneath the cloud of the targeted updraft. This sequence is illustrated in Figure 3.2.



$\mathbf{s}_a$	arrival state	$\mathbf{s}_e$	exit state	v_{si}	aircraft sink rate
V_a	airspeed	V_g	ground speed	γ	flight path angle
z_{min}	minimum altitude	z_{cb}	cloud base altitude	$\mathbf{p}_u$	updraft position
w	climb rate	Δz	gained altitude	$\mathbf{q}_u$	hori. dist. to the updraft

Figure 3.2: Interthermal glide

In the interthermal glide mode, the vehicle glides towards a given updraft position $\mathbf{p}_{u_{xy}} = \begin{bmatrix} \mathbf{p}_{u_x}\, \mathbf{p}_{u_y} \end{bmatrix}^T$ in the horizontal plane. Recall the notation suggested in Chapter 2, where the sub-

scripts u and xy in $\mathbf{p}_{u_{xy}}$ specify the horizontal position vector components of an updraft. Since the glider moves along a straight path, its position $\mathbf{p}$ changes according to

$$\dot{\mathbf{p}} = \begin{bmatrix} \frac{\mathbf{p}_{u_{xy}} - \mathbf{p}_{xy}}{q_u} V_g \\ -v_{si} \end{bmatrix}, \tag{3.2}$$

where V_g and v_{si} are the horizontal speed over ground and the sink rate respectively, the horizontal distance to the updraft is defined by $q_u = ||\mathbf{p}_{u_{xy}} - \mathbf{p}_{xy}||$, and the term $\left(\mathbf{p}_{u_{xy}} - \mathbf{p}_{xy}\right)/q_u$ is the direction vector of the glider in the horizontal plane. Assuming no wind, the relation between the airspeed V_a and the ground speed V_g is defined as illustrated in Figure 3.3 where γ denotes the flight path angle. The sink rate v_{si} is defined positive downwards. For soaring

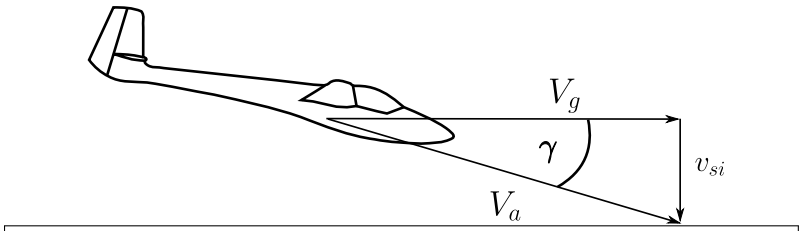


Figure 3.3: Glider kinematics in the vertical plane during *glide*

planes, the sink velocity v_{si} is small, compared to the airspeed V_a. Therefore, both airspeed and ground speed approximately coincide

$$V_g = V_a \cos\gamma \approx V_a \text{ , } \gamma \ll 1. \tag{3.3}$$

In this work, the MacCready airspeed V_{MC}, introduced by the author of [45], is selected as longitudinal guidance command during interthermal glide. It is the optimal airspeed for minimizing the cumulated time of an interthermal glide and its succeeding climb back to the initial altitude at the beginning of the glide-climb-sequence. The derivation of V_{MC} is shortly recalled. Consider the glider-updraft configuration as introduced in Figure 3.2. Assuming the climb rate of the vehicle in the approached updraft w is known, V_{MC} can be derived as follows: The cumulated time t_{cum} of a glide-climb sequence is the sum of the glide time t_g and the climb time t_c to the initial altitude $\mathbf{p}_z$

$$t_{cum} = t_g + t_c = t_g + \frac{\Delta_z}{w} = \underbrace{\frac{q_u}{V_a}}_{t_g} + \underbrace{\frac{q_u}{V_a}\frac{v_{si}}{w}}_{t_c} = \frac{q_u}{V_a}\left(1 + \frac{v_{si}}{w}\right). \tag{3.4}$$

with w specifying the glider's climb rate during thermal climb. The vehicle's polar curve that relates airspeed to sink rate can be stated as

$$v_{si} = aV_a^2 + bV_a + c, \tag{3.5}$$

where a, b, c are the known coefficients. We can substitute the sink rate in Equation (3.4), yielding

$$\begin{aligned} t_g + t_c &= \frac{q_u}{V_a}\left(1 + \frac{aV_a^2 + bV_a + c}{w}\right) \\ &= \frac{bq_u}{w} + \frac{q_u}{V_a w}\left(aV_a^2 + c + w\right). \end{aligned}$$

The optimal airspeed setting is therefore the solution of

$$V_{MC} = \min_{V_a} \frac{q_u}{V_a w}\left(aV_a^2 + c + w\right). \tag{3.6}$$

Differentiating with respect to the airspeed, and solving yields the MacCready airspeed

$$V_{MC} = \sqrt{\frac{c+w}{a}}. \tag{3.7}$$

Note that the MacCready airspeed solely depends on the climb rate and the vehicle's aerodynamic performance. Figure 3.4 illustrates the relation between w and V_{MC} for the polar curve of the glider from Appendix A.

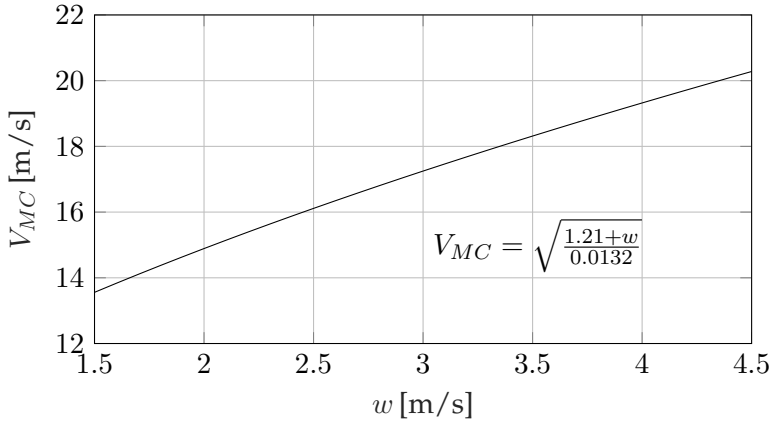


Figure 3.4: MacCready airspeeds as function of the climb rate, calculated for the Cularis UAV

Inserting Equation (3.7) into Equation (3.2), we can predict the arrival state $\mathbf{s}_a$ at the designated updraft from some state $\mathbf{s}$ and the updraft parameters $\mathbf{x}_u = [\mathbf{p}_u \; t_v \; w]^T$ as follows

$$\mathbf{s}_a = f(\mathbf{s}, \mathbf{x}_u) = \mathbf{s} + \begin{bmatrix} \frac{\mathbf{p}_{u_{xy}} - \mathbf{p}_{xy}}{q_u} V_{MC} \\ -v_{si}(V_{MC}) \\ 1 \end{bmatrix} \frac{q_u}{V_{MC}} = \begin{bmatrix} \mathbf{p}_{u_{xy}} \\ \mathbf{p}_{z,a} \\ t_a \end{bmatrix}, \tag{3.8}$$

where the arrival altitude $\mathbf{p}_{z,a}$ can be expressed as function of the arrival time and the Mac-Cready speed setting

$$\mathbf{p}_{z,a} = \mathbf{p}_z - v_{si}(V_{MC})(t_a - t),$$

with

$$t_a = \frac{q_u}{V_{MC}} + t.$$

3.2.2. Thermal Climb

As further illustrated in Figure 3.2, the thermal climb phase is the transition from the arrival state $\mathbf{s}_a$ to the exit state $\mathbf{s}_e$. From the updraft map, we know the average updraft velocity of an updraft $w_{u,rz}$, as introduced in Equation (2.6). The climb rate w does however not only depend on the updraft velocity, but also on the glider's sink rate and its position relative to the updraft center. According to the updraft model from [23] the updraft velocity profile w_u is represented by a bell curve. Therefore, the updraft velocity decreases from the chord line of the updraft outwardly. Assuming the glider to be perfectly centered on a circle aroud the updraft's center where the radius of the circle is small compared to the dimension of the updraft, the vehicle will primarily be impacted by the updraft's core velocity w_p. From [23], the relation between w_p and the average updraft velocity $w_{u,r}$ at a given altitude is

$$w_p = 3 w_{u,r} \frac{r_2^3 - r_2^2 r_1}{r_2^3 - r_1^3}, \tag{3.9}$$

where r_1 and r_2 are the inner and the outer radius of the updraft. As further explained in [23], both radii depend on the altitude and the convective mixing layer thickness z_i. However, for the sake of simplicity, the radii are assumed to remain constant over altitude. For typical convective mixing layer thicknesses of $z_i = 2000\,\text{m}$, the fraction in Equation (3.9) is of the order of 0.7. We can therefore state a simplified relation between the average updraft velocity $w_{u,rz}$

and the average core velocity $w_{p,z}$

$$w_{p,z} = 3 \cdot 0.7\, w_{u,rz}. \tag{3.10}$$

Compensating $w_{p,z}$ for the glider's sink rate during climb, the climb rate w of the glider is stated as

$$w = w_{p,z} - v_{si} \tag{3.11}$$

Then, the position derivative in thermal climb can be defined as

$$\dot{\mathbf{p}} = \begin{bmatrix} 0 \\ 0 \\ w \end{bmatrix}. \tag{3.12}$$

Note that Equation (3.12) is a simplification, serving as computationally efficient motion prediction model. In reality, the glider performs a spirally shaped climb trajectory. The altitude at the exit state can be noted as

$$\mathbf{p}_{z,e} = \mathbf{p}_{z,a} + \Delta z,$$

where Δz is the gained altitude. In this work, we consider two conditions for terminating a climb. First, when the glider reaches the cloud base altitude z_{cb}. Second, when the updraft vanishes. That being said, the gained altitude Δz is defined as

$$\Delta z = \min(t_v - t_a, \frac{z_{cb} - \mathbf{p}_{z,a}}{w})w, \tag{3.13}$$

where t_a stands for the arrival time at the updraft. With Equation (3.13), the exit state $\mathbf{s}_e$ is a function of both the arrival state $\mathbf{s}_a$ and the updraft vector $\mathbf{x}_u$

$$\mathbf{s}_e = f\left(\mathbf{s}_a, \mathbf{x}_u\right) = \mathbf{s}_a + \begin{bmatrix} 0 \\ 0 \\ w \\ 1 \end{bmatrix} \frac{\Delta z}{w}. \tag{3.14}$$

3.2.3. Final Glide

The final glide is the transition between some state $\mathbf{s}$ and a goal state $\mathbf{s}_G$ such that the horizontal position of the glider corresponds to the horizontal position of the destination point, i.e. $\mathbf{p}_{xy} =$

$\mathbf{p}_{D_{xy}}$. Recalling Equation (3.3), the duration t_f of the final glide can be stated as the ratio of the horizontal distance to the destination and the airspeed setting

$$t_f = \frac{q_D}{V_a}. \tag{3.15}$$

As an optimal airspeed for the final glide, we seek to find the airspeed setting $V_a^\star$ that minimizes the flight time t_f of the final glide such that the minimum altitude constraint is not violated. Mathematically, this can be stated as

$$\min_{\substack{V_{a,min} \leq V_a \leq V_{a,max} \\ -v_{si}(V_a)t_f + \mathbf{p}_z > z_{min}}} t_f, \tag{3.16}$$

where $V_{a,min}$ and $V_{a,max}$ define the airspeed envelope. Note that for a glider, t_f is minimized when arriving the lowest at the destination. Therefore, the following holds

$$t_f = \frac{\mathbf{p}_z - z_{min}}{v_{si}} = \frac{\mathbf{p}_z - z_{min}}{aV_a^2 + bV_a + c}, \tag{3.17}$$

where $\mathbf{p}_z$ is the glider's altitude at the beginning of the final glide. Rearranging, the seeked optimal airspeed setting for final glide, $V_a^\star$ is obtained by calculating the zeros of the following quadratic function

$$0 = aV_a^2 + \left(b + \frac{z_{min} - \mathbf{p}_z}{q_D}\right) V_a + c$$

with

$$V_a^\star = -\frac{1}{2a}\left(b + \frac{z_{min} - \mathbf{p}_z}{q_D}\right) + \frac{1}{2a}\sqrt{\left(b + \frac{z_{min} - \mathbf{p}_z}{q_D}\right)^2 - 4ac}. \tag{3.18}$$

Notice that the final glide law can easily be extended to incorporate horizontal wind, as shown in [17]. The airspeed setting $V_a^\star$, as function of the excess altitude $\mathbf{p}_z - z_{min}$ and the remaining horizontal distance to the destination q_D, is illustrated in Figure 3.5 for the Cularis glider model from Appendix A.

Also note that here, $V_a^\star$ is bounded by the airspeed of the best glide $V_{a,min} = 10\,\mathrm{m/s}$ and the maximum airspeed of $V_{a,max} = 25\,\mathrm{m/s}$ respectively. In other terms, if the destination cannot be reached without undercutting the minimum altitude, the selected airspeed for final glide is the airspeed of the best glide. Conversely, a maximum airspeed of 25 m/s is tolerated.

Given the airspeed for the final glide, we can introduce goal states $\mathbf{s}_G \in \left\{\mathbf{s} | \mathbf{p}_{xy} = \mathbf{p}_{D_{xy}}\right\}$. Goal

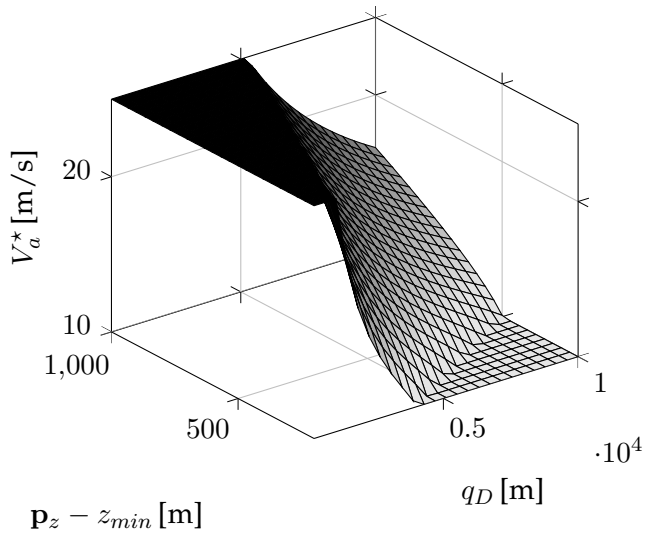


Figure 3.5: $V_a^\star$ as function of the excess altitude $\mathbf{p}_z - z_{min}$ and the horizontal distance q_D to the destination

states are predicted from the vehicle's current state $\mathbf{s}$ and the destination $\mathbf{p}_D$ according to

$$\mathbf{s}_G = f(\mathbf{s}, \mathbf{p}_D) = \mathbf{s} + \begin{bmatrix} \frac{\mathbf{p}_{D_{xy}} - \mathbf{p}_{xy}}{q_D} V_a^\star \\ -v_{si}(V_a^\star) \\ 1 \end{bmatrix} \frac{q_D}{V_a^\star}. \tag{3.19}$$

3.3. Cross-Country Soaring in an Uncertain Environment

3.3.1. Description

Different flight paths have to be compared for identifying the shortest path in time to the target. Therefore, consider Figure 3.6. Suppose the pilot/autopilot has to make a decision: One option is to approach the updraft, regain altitude and then to glide to the destination. The second option is to directly glide to the destination.

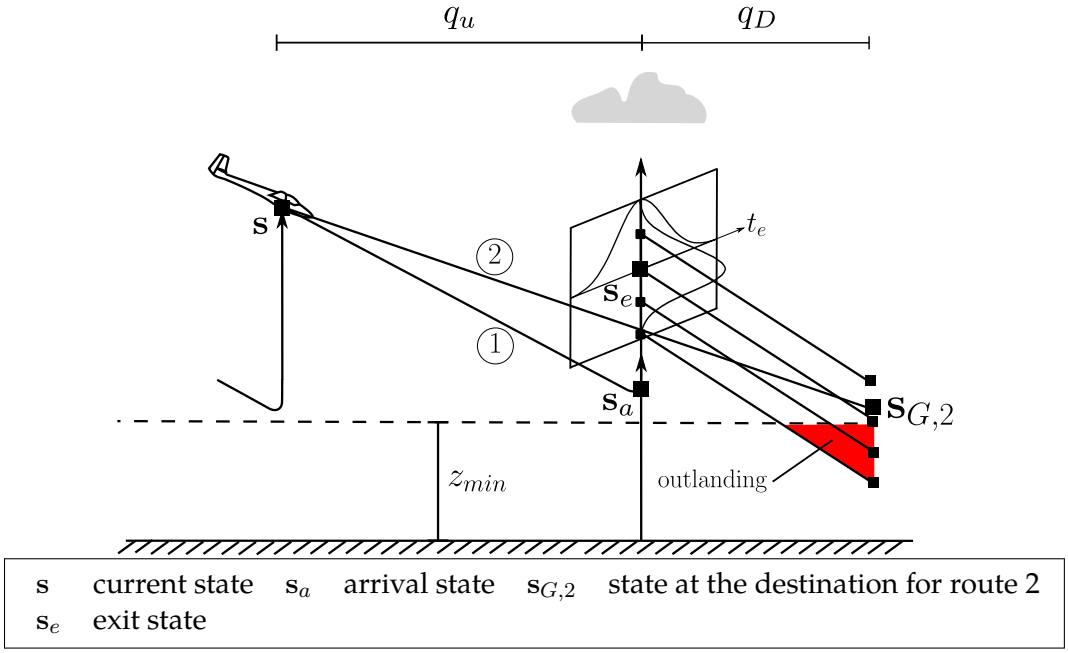


Figure 3.6: The effect of $\mathbf{p}_{z,e}$ on outlandings

Furthermore, suppose the first option ① appears to be faster in time, since the vehicle can approach the updraft with an increased airspeed, regains altitude and initiates a final glide with increased airspeed towards the destination. In contrast, the second option ② is slower in time since no more potential energy is gained on the way to the destination and the vehicle has thus to fly at a lower airspeed for not arriving at the destination below the minimum altitude.
For the decision-making, the vehicle's current state $\mathbf{s}$, the glider's polar curve parameters, and the estimated updraft parameters $\hat{\mathbf{x}}_u = \left[\hat{\mathbf{p}}_u^T \ \hat{t}_v \ \hat{w}\right]^T$ are given, as introduced in Section 2.1. Additionally, the uncertainty which is related to the updraft state is known in terms of the corresponding covariance matrix $\mathbf{P}^{\mathbf{x}_u}$. Given Equations (3.8), (3.14) and (3.19), the pilot/autopilot can determine the state $\mathbf{s}_G$ and evaluate both the cumulated flight time and the arrival altitude for a given flight path. However, all updraft parameters are uncertain, rendering the predicted

state $\mathbf{s}_T$ inherently uncertain. As was shown in Chapter 2, the updraft position tends to converge significantly faster than both the estimates of the vanish time and the updraft velocity. We therefore only consider uncertainties related to these parameters, i.e. the vanish time and updraft velocity. The uncertainties of these parameters enter the state propagation process through Equations (3.13) and (3.14). If we start propagating the vehicle's position from $\mathbf{s}$, the first uncertain state will be $\mathbf{s}_e$. Note that the arrival state $\mathbf{s}_a$ is deterministic since both altitude and time depend on the MacCready airspeed which is known a priori for a given updraft. Also note, that the uncertainties are considered to remain constant over time. As horizontal position uncertainties are not considered, one can separate states $\mathbf{s}$ according to

$$\mathbf{s} = \begin{bmatrix} \mathbf{s}_d \\ \mathbf{s}_p \end{bmatrix}, \tag{3.20}$$

with

$$\begin{aligned} \mathbf{s}_d &= \begin{bmatrix} \mathbf{p}_x \\ \mathbf{p}_y \end{bmatrix} \\ \mathbf{s}_p &= \begin{bmatrix} \mathbf{p}_z \\ t \end{bmatrix}, \end{aligned} \tag{3.21}$$

where the subscripts d and p stand for deterministic and probabilistic respectively. Since the estimation error is considered Gaussian, $\mathbf{s}_p$ is a random vector

$$\mathbf{s}_p \sim \mathcal{N}\left(\bar{\mathbf{s}}_p, \mathbf{P}^{\mathbf{s}_p}\right). \tag{3.22}$$

Reconsidering the example from Figure 3.6 both the altitude and time of arrival at the destination are uncertain when deciding for route ①. In other terms, this particular route might be faster but also bears the risk of an outlanding. The state uncertainty primarily depends on the spatial constellation between the vehicle, the updraft and the destination, as well as the estimation parameters. While the constellation can obviously not be changed, it is possible to integrate the estimation uncertainties in order to thoroughly evaluate whether a decision is beneficial or not. This explains the importance of considering the vanish time and the updraft strength uncertainties in the path planning problem for VBACCS. Integrating these uncertainties, the planning task from Section 3.1 becomes a constrained and time-dependent Stochastic Shortest Path (SSP) problem.

3.3.2. A Numerical Example

Suppose the following inputs are given to the simple decision-making process from Figure 3.6:

$$
\begin{aligned}
\mathbf{s} &= \begin{bmatrix} 0\,\mathrm{m} & 0\,\mathrm{m} & 1000\,\mathrm{m} & 0\,\mathrm{s} \end{bmatrix}^T, \\
\bar{\mathbf{x}}_u &= \begin{bmatrix} 4500\,\mathrm{m} & 0\,\mathrm{m} & 1200\,\mathrm{m} & 400\,\mathrm{s} & 3\,\mathrm{m/s} \end{bmatrix}^T, \\
\mathbf{P}^{\mathbf{x}_u} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & (200\mathrm{s})^2 & 0 \\ 0 & 0 & 0 & 0 & (0.4\mathrm{m/s})^2 \end{bmatrix}, \\
\mathbf{p}_{D_{xy}} &= \begin{bmatrix} 7700\,\mathrm{m} & 0\,\mathrm{m} \end{bmatrix}^T, \\
z_{\text{min}} &= 500\,\mathrm{m},
\end{aligned}
$$

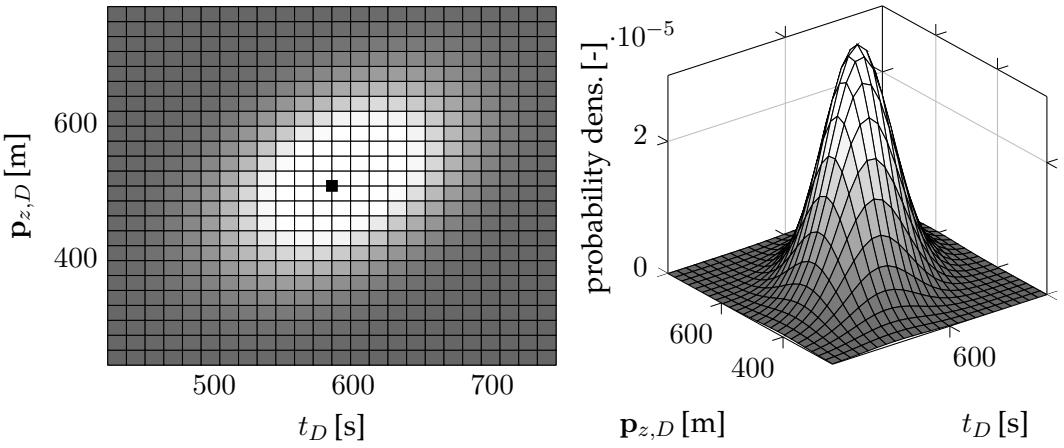
where the updraft state is supposed to be Gaussian, such that $\mathbf{x}_u \sim \mathcal{N}(\bar{\mathbf{x}}_u, \mathbf{P}^{\mathbf{x}_u})$. Given the initial state $\mathbf{s}$, and the uncertain updraft parameters, we can determine the vehicle's state $\mathbf{s}_G$, by applying an Uncertain Transformation (UT). Thereby, we propagate the vehicle's state through Equations (3.8), (3.14) and (3.19), and recapture the uncertainty in terms of

$$
\bar{\mathbf{s}}_G = \begin{bmatrix} \underbrace{7700\,\mathrm{m} & 0\,\mathrm{m}}_{\mathbf{s}_p} & \underbrace{502\,\mathrm{m} & 592\,\mathrm{s}}_{\mathbf{s}_p} \end{bmatrix}^T, \tag{3.23}
$$

where the state covariance of the probabilistic state component $\mathbf{s}_p$ is given by

$$
\mathbf{P}^{\mathbf{s}_{G_p}} = 1000 \begin{bmatrix} 7.90\,\mathrm{m}^2 & 1.89\,\mathrm{ms} \\ 1.89\,\mathrm{ms} & 2.89\,\mathrm{s}^2 \end{bmatrix}.
$$

That is, both the altitude and the time at the target have standard deviations of approximately 89 m and 54 s respectively. The Probability Density Function (PDF) of the uncertain destination state is illustrated in Figure 3.7. Note that the mean of the predicted altitude at the target is approximately 502 m, i.e. slightly above the minimum altitude constraint of 500 m (see Figure 3.7a). Thus, a purely deterministic path planner would judge this path feasible. However, we can calculate the probability of an outlanding by integrating over the bivariate PDF of the



(a) Top view (b) Side view

Figure 3.7: Probability density of the uncertain terminal state

uncertain state. Let the bivariate PDF be noted as

$$f_{\mathbf{s}_{G_p}} = \frac{1}{2\pi|\mathbf{P}^{\mathbf{s}_{G_p}}|^{\frac{1}{2}}} \mathrm{e}^{-\frac{1}{2}\left(\mathbf{s}_{G_p}-\bar{\mathbf{s}}_{G_p}\right)^T {\mathbf{P}^{\mathbf{s}_{G_p}}}^{-1}\left(\mathbf{s}_{G_p}-\bar{\mathbf{s}}_{G_p}\right)}, \tag{3.24}$$

with $|\cdot|$ standing for the determinant of a matrix. The probability of an outlanding is the result of the following integral

$$P\left(\mathbf{p}_{z,D} < z_{min}\right) = \int_{-\infty}^{\infty}\int_{-\infty}^{z_{min}} f_{\mathbf{s}_{G_p}} d\mathbf{p}_{z,D} dt = 0.47. \tag{3.25}$$

Thus, on the regarded example, the vehicle has a 47% risk of of an outlanding when deciding to approach the destination via the designated cloud. In contrast, by means of Equation (3.19), the flight time of a direct final glide is 611 s, i.e. approximately 19 s slower, however without the risk of an outlanding.

3.4. Recall on Markov Decision Processes and Dynamic Programming

3.4.1. Context

In robotics, many contributions with focus on path planning consider deterministic motion processes [48, 65, 66, 67, 68]. However, uncertainties occur in nearly all robotic applications and can sensitively impact the motion. Most commonly, sources of uncertainty include exogenous disturbances, imperfect perception of both the vehicle's state and the environment, and model uncertainties. As a consequence, control inputs result in stochastic outcomes, calling for decision-making methods that consider the process' uncertainty.

Decision-making in uncertain environments is a broad domain including various fields, one of these being the calculation of shortest paths under motion uncertainty. No matter the field, decision-making under uncertainty relies on methods for modeling the uncertain process. A particularly powerful framework dedicated to this task are Markov Decision Processes (MDPs). MDPs are sequential decision models and were introduced by the authors of [69] and [70]. According to the authors of [71], the broad application of MDPs ranges from finance, over agriculture, to inventory and production. With a trend toward more efficient computer processors, MDPs have also gained the attention of researchers in robotics. For an example, the authors of [72] perform autonomous inverted helicopter flight, relying on reinforcement learning and MDPs. Others have relied on MDPs for UAV flocking [73] or target following [74]. A work with similarities to the VBACCS problem is provided by the authors of [75], where the framework of MDPs is utilized for UAV energy harvesting in horizontal winds.

For systems with a large number of states and infinite planning horizons, MDPs tend to be computationally expensive. If the system's uncertainty is known in advance, the decision-making problem can be solved off-line where the calculation time is not predominant. However, for VBACCS, there is no way of quantifying the uncertainty beforehand. This is because updrafts occur randomly over the region the vehicle operates in, which is an unpredictable process. Therefore, the decision-making problem has to be solved in real-time, and preferably onboard the vehicle. As compared to many other applications where the problem size has to be reduced, in VBACCS, both the comparably small problem size and a naturally limited horizon allow a near-optimal onboard solution calculation in short time. That said, the MDP framework is adopted for tackling the VBACCS planning problem.

In the following, a special class of MDPs, namely Stochastic Shortest Path MDPs [76] will be recalled which is particularly suited to model the VBACCS planning problem.

3.4.2. Stochastic Shortest Path Markov Decision Processes

This work relies on the SSPMDP definition, proposed by the author of [76]. The SSPMDP is represented by a tuple $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{C}, \mathcal{G}, s_0, \mathcal{K} \rangle$. The components of the tuple are defined as:

- $\mathcal{S}$ indicates the set of all possible discrete system states.
- $\mathcal{A}$ is a finite set of discrete actions that the agent can execute. $A(s) \subseteq \mathcal{A}$ defines the subset of applicable actions at s. That is, $A(s) = \{\forall a \in \mathcal{A} | I(s,a) \text{ is true } \}$ where $I(s,a)$ is true when a is applicable at s and false otherwise.
- $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S}$ stands for the stochastic transition model where $\mathcal{P}(s'|s,a)$ is the probability of transitioning from state s to state s' when executing the action a in state s.
- $\mathcal{C} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, \infty)$ is the cost model such that $C(s'|s,a)$ defines the cost of transitioning from s to s' when executing action a in state s. $C(s'|s,a)$ is strictly positive whenever s is not a goal state.
- $\mathcal{G} \subseteq \mathcal{S}$ defines the set of all goal states s_G, also known as absorbing states, such that for all $s_G \in \mathcal{G}$, for all $a \in \mathcal{A}$, and for all $s' \notin \mathcal{G}$ the following holds: $\mathcal{P}(s' \notin \mathcal{G} | s_G, a) = 0$, and $\mathcal{C}(s_G | s_G, a) = 0$. Reaching any of the states in $\mathcal{G}$ terminates the execution.
- s_0 is the initial state.
- $\mathcal{K}$ defines a finite set of decision epochs such that $\mathcal{K} \coloneqq \{0, \ldots, N\}$. Elements of $\mathcal{K}$ specify the current decision epoch with running index k. Throughout this thesis, a convention is adopted where the last decision is made at the epoch $N - 1$.

In comparison to the more general case of MDPs, the SSPMDP features some major particularities. It deals with costs and not with rewards. Therefore, the problem is a minimization and not a maximization problem. Additionally, SSPMDPs contain absorbing goal states, i.e. states where actions have no effect on the cost/reward.

3.4.3. The MDP related Problem for Finite Horizons

The problem which is related to a SSPMDP consists of finding a policy, i.e. a sequence of actions $\pi = (\pi_0, \pi_1, ...\pi_{N-1})$, that enables to bring the agent from some current state to a goal state, while minimizing a given criterion. Note that the element $\pi_k(s_k)$ of a policy specifies the action to be taken in node s at decision epoch k.
Let the value function V represent a metric for the utility of a policy. More specifically, V^π specifies the expected total transition cost under some policy π, when starting in a state s_0 in the first decision epoch.

$$V^\pi(s) = \mathbb{E}\left[\sum_{k=0}^{N-1} \mathcal{C}\left(s_{k+1}|s_k, \pi_k(s_k)\right)\middle| s_0 = s\right]. \tag{3.26}$$

The problem of finding the optimal policy $\pi^\star$ can then be stated such that

$$\pi^\star = \min_\pi V^\pi(s). \tag{3.27}$$

Different policies have to be evaluated and compared against each other, in order to identify the optimal policy $\pi^\star$. However, it is not trivial to calculate the cumulated expected cost from Equation (3.26) in a forward way. Instead, recursion can be applied to determine the expected cost over $N - k$ remaining decision epochs

$$V_k^\pi(s) = \sum_{s' \in \mathcal{S}} \mathcal{P}\left(s'|s, \pi(s)\right)\left[\mathcal{C}\left(s'|s, \pi(s)\right) + V^\pi(s')\right]. \tag{3.28}$$

3.4.4. Backward Induction for Solving SSPMDPs

In the following, all values V_N corresponding to all states of the set $\mathcal{S}_N \in \mathcal{S}$, i.e. states at decision epoch N are supposed to be given. Since we seek to identify the next action to execute for a state with $N-1$ remaining actions to take, we can apply the recursion from Equation (3.28) to project the values of the states $\mathcal{S}_N$ back in time, i.e. in reversed chronological order while minimizing the expected future cost and extracting the corresponding policy. The underlying idea is the Bellman principle. It states that an optimal policy has the property that whatever the initial state and decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision.
Starting in decision epoch $k = N - 1$ and successively decrementing k by one, the value of all

states is minimized according to

$$V_k^\star(s) \leftarrow \min_{a\in\mathcal{A}} \sum_{s'\in\mathcal{S}} \mathcal{P}(s'|s,a)\left[C(s'|s,a) + V_{k+1}^\star(s')\right]. \tag{3.29}$$

The identification of the optimal values $V_k^\star$ leads to the corresponding optimal policy $\pi^\star$ which is defined as

$$\pi_k^\star(s) = \arg\min_{a\in A} \sum_{s'\in\mathcal{S}} \mathcal{P}(s'|s,a)\left[C(s'|s,a) + V_{k+1}^\star(s')\right]. \tag{3.30}$$

Due to the reversed chronological optimization, this procedure is known as backward induction [77, 78]. The pseudo code for the backward induction procedure is noted in Algorithm 1. The asymptotic time complexity (noted O) of the backwards induction algorithm depends on both the number of the backups and the number of the states. The complexity of one backup is $O\left(|\mathcal{S}|^2|\mathcal{A}|\right)$ where $|\cdot|$ represents the number of elements contained in the respective set. Thus, the time complexity for a horizon-N backward induction is $O\left(N|\mathcal{S}|^2|\mathcal{A}|\right)$.

Note that backwards induction can also be thought of as a version of the more widely known value iteration algorithm [56], applied to finite horizon processes.

Algorithm 1: Backwards Induction - `backwardsinduction()`

Input : MDP with $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{C}, \mathcal{K})$
Output: $V^\star(s)$ min. value
$\pi^\star(s)$ optimal policy

Initialize: $V_N^\star(s_N) = 0$
for $k = N-1$ **to** $k = 0$ **do**
  for each non-goal state $s \in \mathcal{S}_k$ **do**
    $V_k^\star(s) \leftarrow \min_{a\in\mathcal{A}} \sum_{s'\in\mathcal{S}} \mathcal{P}(s'|s,\pi(s))\left[\mathcal{C}(s'|s,\pi(s)) + V_{k+1}^\star(s')\right]$
    $\pi_k^\star(s) \leftarrow \arg\min_{a\in\mathcal{A}} \sum_{s'\in\mathcal{S}} \mathcal{P}(s'|s,\pi(s))\left[\mathcal{C}(s'|s,\pi(s)) + V_{k+1}^\star(s')\right]$
  end
  k = k-1
end

3.5. Cross-Country Soaring as a Markov Decision Problem

3.5.1. MDP State Definition

For the cross-country soaring problem, the set of states $\mathcal{S}$, as defined in Section 3.4.2, contains all possible 4 D position-time states s, noted as

$$s = (\mathbf{p}_x, \mathbf{p}_y, \mathbf{p}_z, t)\,. \tag{3.31}$$

3.5.2. Definition of Actions

In this work, two different types of actions are considered. These are actions of type a_α*: transition to thermal* and actions of type a_β*: transition to target*. The first action is defined by a sequence of an interthermal glide (Section 3.2.1), followed by a thermal climb (see Section 3.2.2 for an explanation). The applicability function for actions a_α is defined such that

$$I\,(s,a)\,|a \in \mathcal{A}_\alpha = \begin{cases} \text{true, if } \mathbf{p}_{z,a} \geq z_{min} \text{ and } |\Delta\Psi| \leq 0.5\pi \\ \text{false, otherwise,} \end{cases} \tag{3.32}$$

where $\mathcal{A}_\alpha$ is the set of all actions a_α. Recall that $\mathbf{p}_{z,a}$ is the glider's altitude when arriving at the designated updraft. That is, following Equation (3.32), an updraft can be approached on condition that the glider reaches the updraft position without violating the minimum altitude constraint. Furthermore, course deviations $\Delta\Psi$ w.r.t. the target are limited to $\pm 90°$. Subsequently, the glider can only approach updrafts that lie in the sector between its current position and the target.

In contrast to *transition-to-thermal* actions, *transition-to-target* actions are defined as always applicable

$$I\,(s,a)\,|a \in \mathcal{A}_\beta = \text{true}, \tag{3.33}$$

where the set $\mathcal{A}_\beta$ contains all actions of type a_β. Though the applicability for *transit-to-target* actions is true from any state, this does not mean that the glider will reach the target without undercutting the minimum altitude constraint from any state.

3.5.3. State Transition Model

In this work, it is explicitly distinguished between the deterministic and the probabilistic part of the state transition model. The deterministic part relates mean states to mean outcomes over actions. The probabilistic part of the transition model deals with the calculation of discrete state transition probabilities.
We recall, that the framework of MDPs can also be used to calculate a deterministic planning solution. To do so, it suffices to ignore the stochastic part of the transition model. This is done in the evaluation of this chapter, see Section 3.6, in order to illustrate the benefits of probabilistic path planning in VBACCS, compared to deterministic path planning.

Deterministic State Transition Model

The transition model for a_α actions takes a state s as input and predicts an exit state s_e based upon the glider motion model from Section 3.2.

$$s_e = \begin{pmatrix} \mathbf{p}_{u,xy} \\ \mathbf{p}_z - v_{si}\left(V_{MC}\right)\frac{q_u}{V_{MC}} + \Delta_z \\ t + \frac{q_u}{V_{MC}} + \frac{\Delta z}{w} \end{pmatrix}. \tag{3.34}$$

In the sequel, we will define the set of goal states $\mathcal{S}_G = \{\mathcal{S}|\mathbf{p}_x = \mathbf{p}_{D_x}, \mathbf{p}_y = \mathbf{p}_{D_y}\}$. Transition to target actions a_β are then defined to relate states s to goal states $s_G \in \mathcal{S}_G$ according to

$$s_G = \begin{pmatrix} \mathbf{p}_{D_{xy}} \\ \mathbf{p}_z - v_{si}\frac{q_D}{V_{a^\star}} \\ t + \frac{q_D}{V_a^\star} \end{pmatrix}. \tag{3.35}$$

Probabilistic State Transition Model

Uncertain estimates of both the vanish time t_v and the climb rate w propagate into uncertain exit states through Equation (3.34). The probabilistic state transition model quantifies outcomes of actions in terms of probabilities and related conditional expectations.
An Unscented Transform (UT) (see Section 2.4.1) is used to propagate the uncertainty $\mathbf{P}^{\mathbf{x}_u}$ into an uncertain exit state $\mathbf{s}_{e_p}$ and its corresponding covariance $\mathbf{P}^{\mathbf{s}_{e_p}}$ through the system equations Equation (3.34). Two constraints are imposed on the exit altitude:

The first constraint refers to the maximum operational altitude of the glider which we limit by the cloud base altitude z_{cb}. That is, as soon as the glider reaches the cloud base altitude, any climb is interrupted. The second constraint states that the glider does not climb in updrafts with non-positive updraft velocities. Therefore, the exit altitude is inherently equal to or greater than the arrival altitude $\mathbf{p}_{z,a}$ and the exit time is equal or greater than the arrival time. The two constraints are summarized as

$$\begin{aligned} &\mathbf{p}_{z,a} \leq \mathbf{p}_{z,e} \leq z_{cb}, \\ &t_e \geq t_a. \end{aligned} \tag{3.36}$$

For defining a region of possible exit states related to actions a_α, the rectangular bounding box (consisting of exit altitude bounds $\mathbf{p}_{z,\underline{e}}, \mathbf{p}_{z,\bar{e}}$ and exit time bounds $t_{\underline{e}}, t_{\bar{e}}$) is calculated from the normal distribution $\mathbf{s}_{e_p} \sim \mathcal{N}\left(\bar{\mathbf{s}}_{e_p}, \mathbf{P}^{\mathbf{s}_{e_p}}\right)$ and some selected confidence interval. The mathematical derivation of the bounding box is shown in Appendix C. It is important to note that since the state $\mathbf{s}_{e_p}$ features a normal distribution, the ellipse and thus the bounding box can violate the state constraint from Equation (3.36) by comprising regions of the state space that are not accessible. This effect is illustrated in Figure 3.8. Therefore, we saturate the bounding

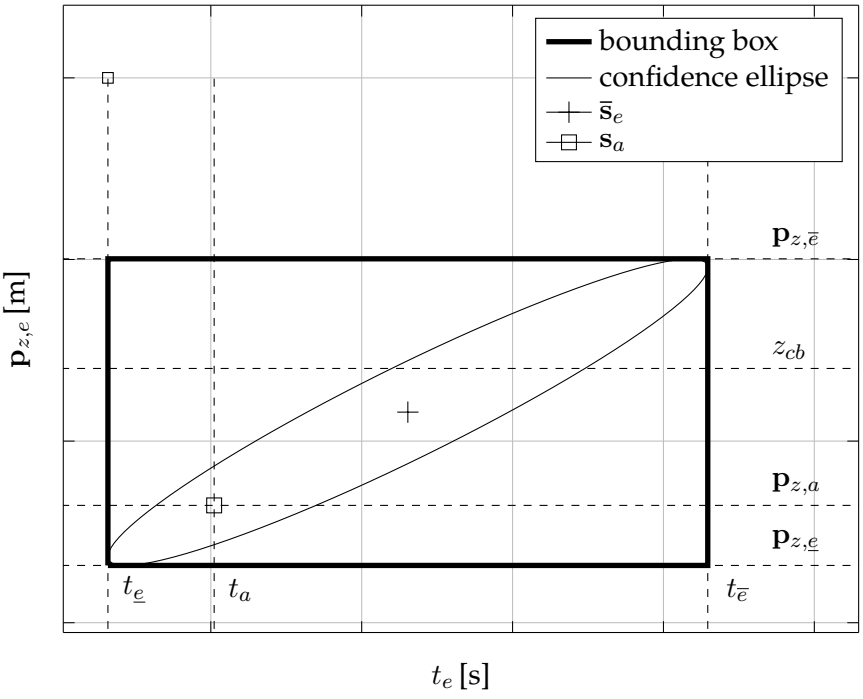


Figure 3.8: Confidence ellipse of $\mathbf{s}_{e_p}$ with related bounding box

box with the constraints, yielding a saturated bounding box

$$\mathbf{p}_{z,\bar{e}} = \min\left(\mathbf{p}_{z,\bar{e}}, z_{cb}\right),$$
$$\mathbf{p}_{z,\underset{\sim}{e}} = \max\left(\mathbf{p}_{z,\underline{e}}, \mathbf{p}_{z,a}\right),$$
$$t_{\underset{\sim}{e}} = \max\left(t_{\underline{e}}, t_a\right),$$
$$t_{\bar{e}} = t_{\bar{e}}.$$

The region comprised by the saturated bounding box is divided in $m \times n$ rectangular bins B, where m and n stand for the number of altitude and time segments respectively. A bin is defined such that

$$B_{ij} \coloneqq \{(\mathbf{p}_{z,e}, t_e) \,|\mathbf{p}_{z,e_{min_i}} \leq \mathbf{p}_{z,e} \leq \mathbf{p}_{z,e_{max_i}}, t_{e_{min_j}} \leq t_e \leq t_{e_{max_j}}\}, \tag{3.37}$$

where $1 \leq i \leq m$, $1 \leq j \leq n$ and $t_{e_{min}}, t_{e_{max}}, \mathbf{p}_{z,e_{min}}, \mathbf{p}_{z,e_{max}}$ are the borders of a bin. The discretized saturated bounding box is illustrated in Figure 3.9 for $m = 4$ and $n = 3$ respectively.

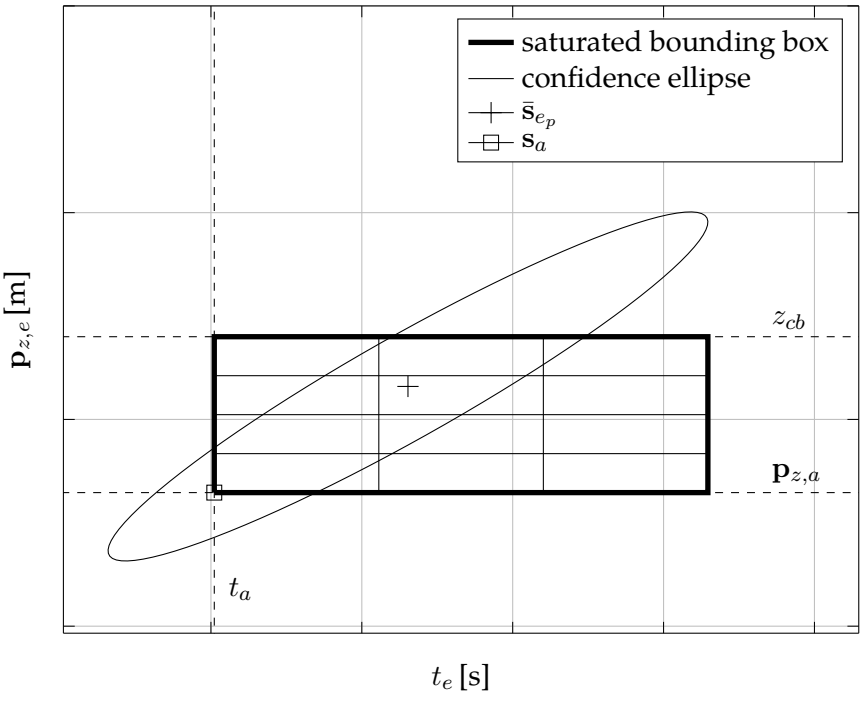


Figure 3.9: Discretized saturated bounding box

The probability of terminating an action from a bin is defined as the normalized integral of the Probability Density Function (PDF) $f_{\mathbf{s}_{e_p}}$ over a region R

$$P\left(\mathbf{s}_{e_p} \in B\right) \coloneqq \frac{1}{\sum_{i=1}^{m \times n} \int_{R_i} f_{\mathbf{s}_{e_p}} d\mathbf{s}_{e_p}} \int_R f_{\mathbf{s}_{e_p}} d\mathbf{s}_{e_p}, \tag{3.38}$$

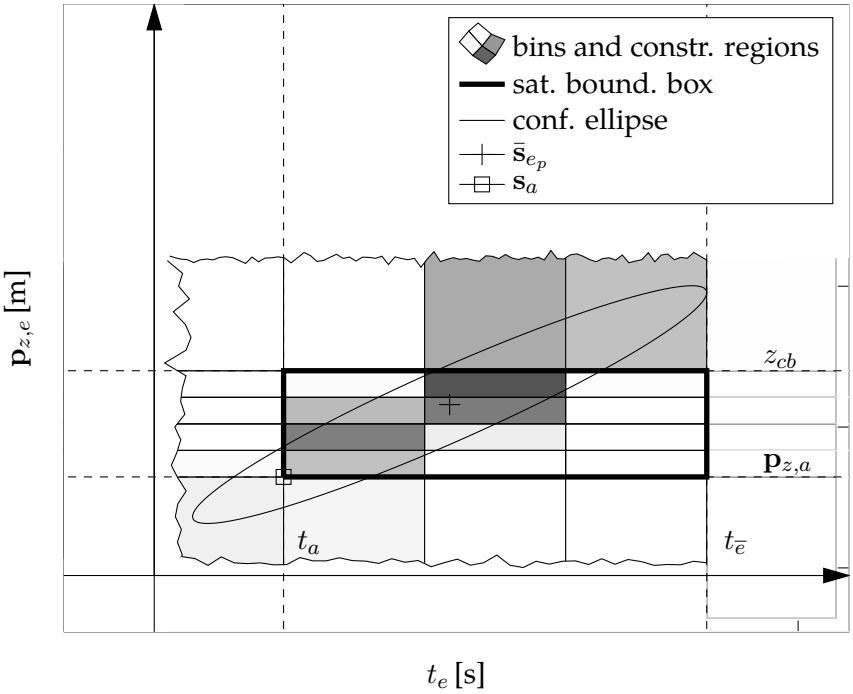
where the integration region R is defined to not only depend on the area B of the related bin. In particular, we define the region as

$$R = \begin{cases} \{(\mathbf{p}_{z,e}, t_e) \,|\mathbf{p}_{z,e} \in [\mathbf{p}_{z,e_{min}}, \infty), t_e \in [t_{e_{min}}, t_{e_{max}}]\}, \text{if } (\mathbf{p}_{z,e_{max}} = z_{cb}, t_{e_{min}} > t_a) \\ \{(\mathbf{p}_{z,e}, t_e) \,|\mathbf{p}_{z,e} \in [\mathbf{p}_{z,e_{min}}, \infty), t_e \in (-\infty, t_{e_{max}}]\}, \text{if } (\mathbf{p}_{z,e_{max}} = z_{cb}, t_{e_{min}} = t_a) \\ \{(\mathbf{p}_{z,e}, t_e) \,|\mathbf{p}_{z,e} \in [\mathbf{p}_{z,e_{min}}, \mathbf{p}_{z,e_{max}}], t_e \in (-\infty, t_{e_{max}}]\}, \text{if} \\ (\mathbf{p}_{z,e_{max}} \leq z_{cb}, \mathbf{p}_{z,e_{min}} > \mathbf{p}_{z,a}, t_{e_{min}} = t_a) \\ \{(\mathbf{p}_{z,e}, t_e) \,|\mathbf{p}_{z,e} \in (-\infty, \mathbf{p}_{z,e_{max}}], t_e \in (-\infty, t_{e_{max}}]\}, \text{if } (\mathbf{p}_{z,e_{min}} = \mathbf{p}_{z,a}, t_{e_{min}} = t_a) \\ \{(\mathbf{p}_{z,e}, t_e) \,|\mathbf{p}_{z,e} \in (-\infty, \mathbf{p}_{z,e_{max}}], t_e \in [t_{e_{min}}, t_{e_{max}}]\}, \text{if } (\mathbf{p}_{z,e_{min}} = \mathbf{p}_{z,a}, t_{e_{min}} > t_a) \\ B, \text{otherwise.} \end{cases} \tag{3.39}$$

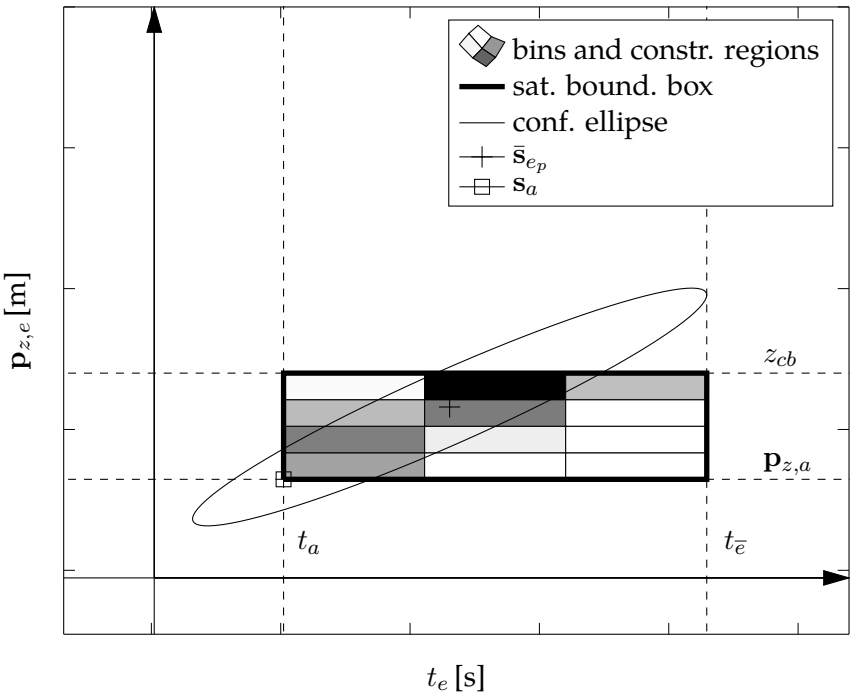
Thereby, if a bin is adjacent to parts of the constrained state region, all probabilities of terminating the action from the adjacent constrained region are mapped onto the bin.

The composition of the bin-related probabilities is explained in Figure 3.10. The first colormap, i.e. Figure 3.10a, illustrates the probabilities, obtained when only integrating over the area. Note that the frayed border marks the outer bound of the constrained region. Figure 3.10b illustrates the bin-related probabilities following Equations (3.38) and (3.39). Consider the upper right bin for an example. According to Figure 3.10a, the probability of terminating an action from this particular bin is approximately zero, when not considering its adjacent regions. However, following Equation (3.39), the probabilities of its adjacent regions are mapped onto it, which increases its related probability. The bin's colour shifts from white to light grey. In the same way, the probability of terminating an action from the upper center bin increases, its colour shifts from dark grey to black.

Note that there is no closed-form solution for the integral of the bivariate Gaussian distribution Equation (3.38). Rather than solving the integral iteratively, the approximation for the bivariate case, as suggested by the authors of [79], is adopted. It constitutes a computationally efficient and yet precise solution.



(a) Unconstrained



(b) Constrained, follwing Equations (3.38) and (3.39)

Figure 3.10: Bin-related probabilities

In addition to the bin-related probability, we seek to calculate the conditional expectation of a bin. Mathematically, it can be stated as

$$\mathbb{E}\left(\mathbf{s}_{e_p}|\mathbf{s}_{e_p} \in B\right) = \frac{1}{P\left(\mathbf{s}_{e_p} \in B\right)} \int_R g\left(\mathbf{s}_{e_p}\right) f_{\mathbf{s}_{e_p}} d\mathbf{s}_{e_p} \tag{3.40}$$

where the function g saturates the state $\mathbf{s}_{e_p}$ such that

$$\mathbf{p}_{z,e,sat} = \begin{cases} z_{cb}, \text{ if } \mathbf{p}_{z,e} \geq z_{cb} \\ \mathbf{p}_{z,a}, \text{ if } \mathbf{p}_{z,e} \leq \mathbf{p}_{z,a} \\ \mathbf{p}_{z,e}, \text{ otherwise.} \end{cases}$$

$$t_{e,sat} = \begin{cases} t_a, \text{ if } t_e \leq t_a \\ t_e, \text{ otherwise.} \end{cases}$$

Thereby, the saturation function g maps all states $\mathbf{s}_{e_p}$ that violate the state constraints onto the nearest non-constraint state regions. In Appendix D it is shown how the integral Equation (3.40) can be solved analytically for rectangular regions, thus again being a computationally efficient approach in comparison to an iterative solution calculation.

The discretized state with both bin-related probabilities and conditional expectations is illustrated in Figure 3.11. As expected, the bin-conditional expectations are not centered in the bin. In addition, Figure 3.12 symbolically illustrates the discrete outcome states of an action and its related probabilities (from the side view).

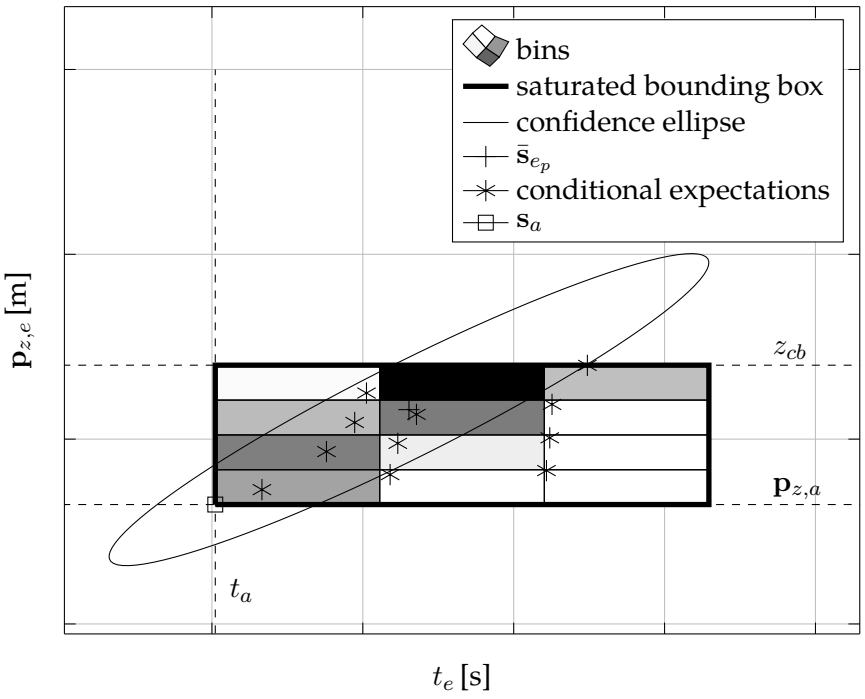


Figure 3.11: Discrete exit states with related bin probabilities

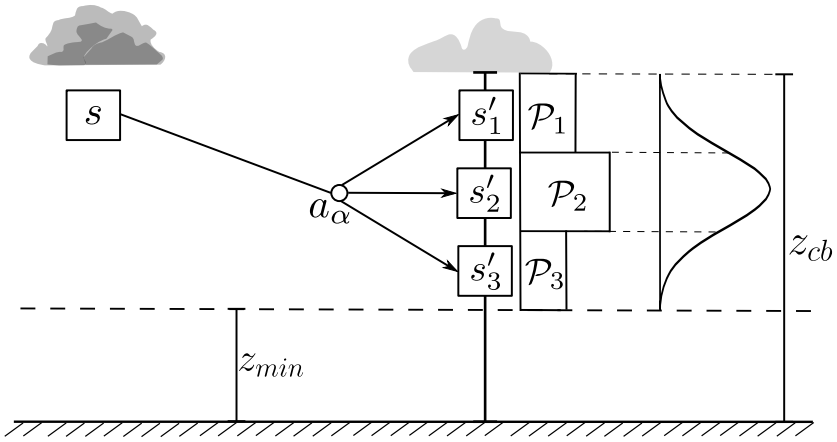


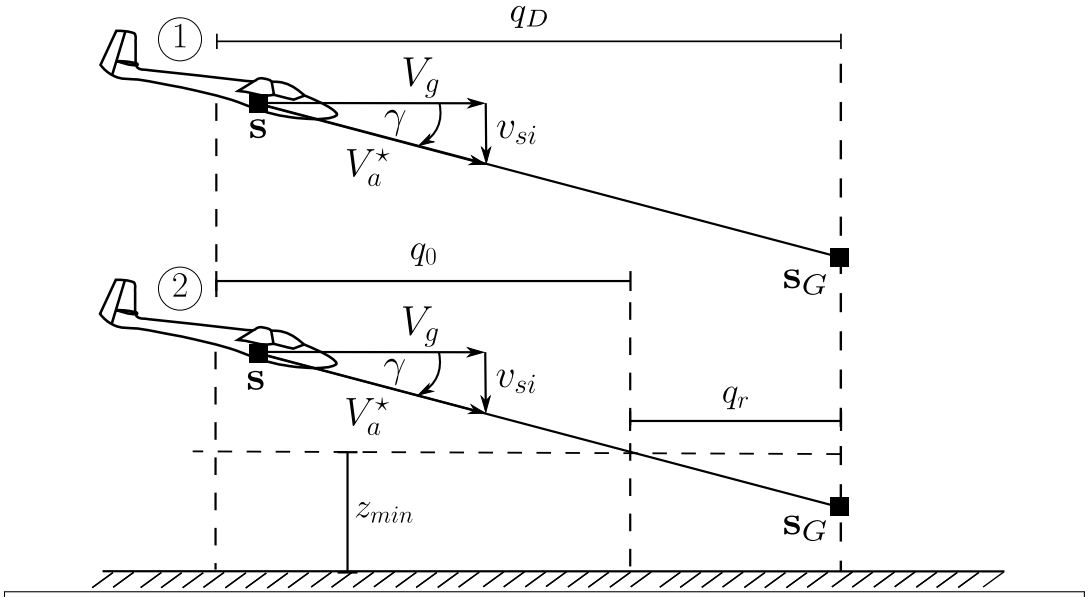
Figure 3.12: Side view of the discretized exit states

3.5.4. Cost Model

A cost model $\mathcal{C}$ is required for defining state transition costs. In the autonomous cross-country soaring problem, the transition cost for actions a_α is the sum of the glide time and the climb time to the respective altitude. Recall that this quantity can be calculated following Equation (3.4). Thus, we can state

$$\mathcal{C}(s'|s, a_\alpha) = t_g + t_c = \frac{q_u}{V_g} + \frac{\Delta z}{w}. \tag{3.41}$$

The cost calculation for transitioning to the target, i.e. actions a_β is illustrated in Figure 3.13 where $V_a^\star$ stands for the constant airspeed during final glide.



$\mathbf{s}$	current state	$\mathbf{s}_G$	goal state	γ	flight path angle
V_a	airspeed	V_g	ground speed	v_{si}	sink rate
z_{min}	minimum alt.	q_0	h. dist. to outland.	q_r	remaining h. dist.
q_D	h. dist. to destination				

Figure 3.13: Cost calculation for *transit-to-target* actions

In case the destination can be reached without violating the minimum altitude constraint (see ① in Figure 3.13), the cost is defined as the glide time.
In case the target cannot be reached without undercutting the minimum altitude z_{min} (see ② in Figure 3.13), the cost is defined as the glide time to the outlanding site plus a penalty Ψ.

For the two cases, the transition cost is summarized as

$$\mathcal{C}(s'|s, a_\beta) = \begin{cases} \frac{q_D}{V_a^\star} & \text{, if } \mathbf{p}_{z,D} \geq z_{min} \\ \frac{q_0}{V_a^\star} + \Psi & \text{, otherwise,} \end{cases} \tag{3.42}$$

where the corresponding distances are

$$q_D = ||\mathbf{p}_{D_{xy}} - \mathbf{p}_{xy}||, \tag{3.43}$$

$$q_0 = \frac{\mathbf{p}_z - z_{min}}{-v_{si}(V_a^\star)} V_a^\star, \tag{3.44}$$

$$q_r = q_D - q_0. \tag{3.45}$$

The penalty Ψ for an outlanding contains both a constant term k_0 and a term that depends on the remaining distance

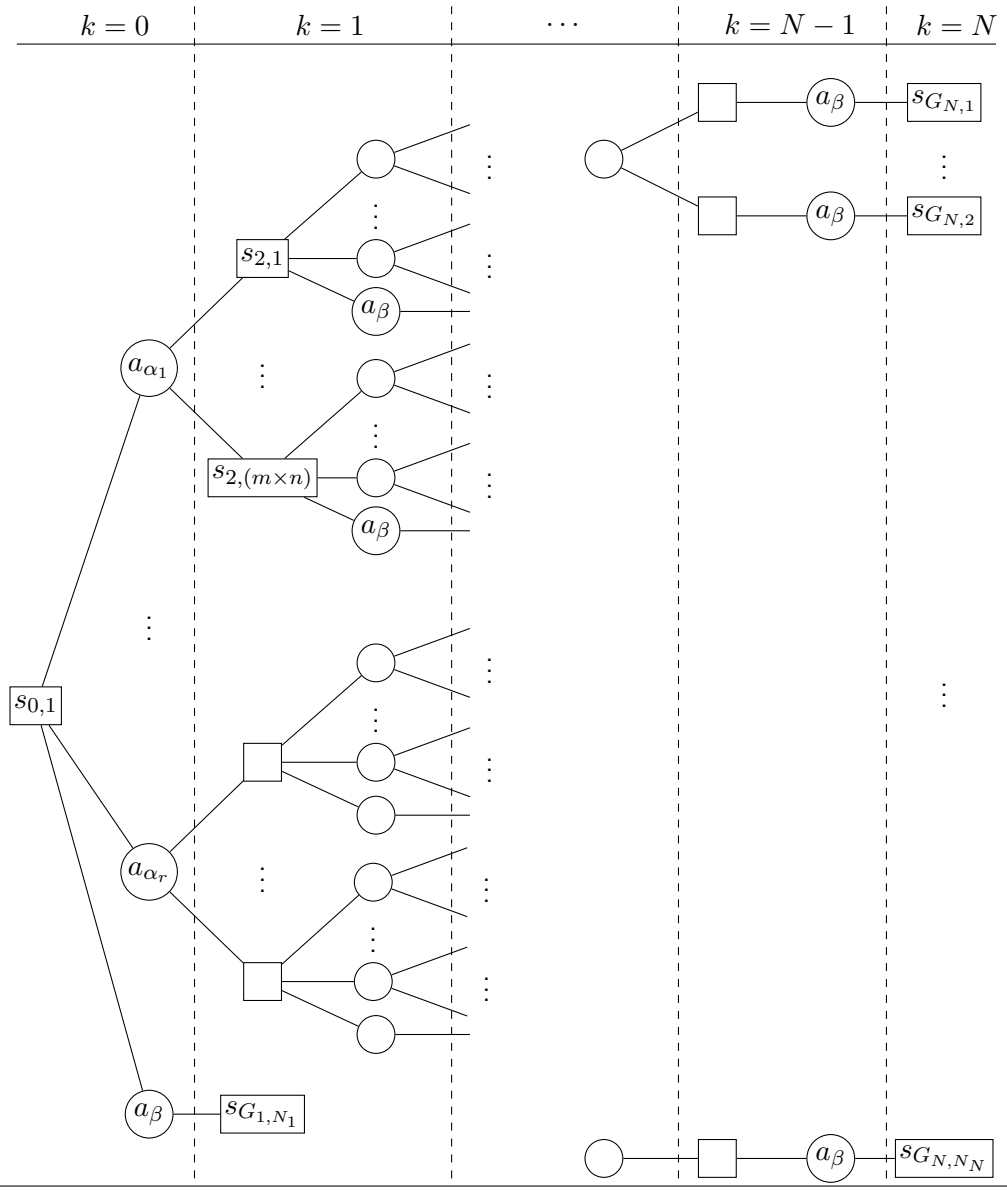
$$\Psi = \frac{q_r}{k_1 V_a^\star} + k_0. \tag{3.46}$$

The dimensionless parameter k_1 serves to scale the constant final glide velocity $V_a^\star$. Recall that $V_a^\star$ corresponds to the best airspeed of maximum lift-to-drag, for the case the glider cannot reach the destination. The term $k_1 V_a^\star$ in Equation (3.46) can be thought of as the velocity of a fictitious conveyor belt that transports the vehicle from the outlanding site to the destination. Then, the first term of Equation (3.46) stands for the transportation time it takes to bring the vehicle from the outlanding site to the destination. The second term, i.e. the time k_0 encapsulates the glider competition rule, that landing out one meter short of the target is always worse than accomplishing the mission without an outlanding [46].

Integrating the notion of an outlanding into the cost is fundamental for the purpose of this thesis. Through the penalty for an outlanding, we can balance between the risk of a mission failure and the performance in terms of flight time.

3.5.5. Tree Construction for Policy Optimization

Given the problem in MDP form, a decision tree featuring $N + 1$ layers is generated, as illustrated in Figure 3.14 where the rectangles/squares define states and the circles specify actions. The tree is constructed in chronological order, starting from the vehicle's current state s_0. At each node in the layers $0 \leq k < N$, the applicability function outputs *transit-to-thermal* actions a_α for the thermals that lie in the glide range and the *transit-to-target* action a_β. As an additional condition, the vehicle can harvest energy only once from each updraft. That is, after having



$s_{i,j}$	state j in layer i	N_i	num. states in layer i	$r+1$	num. actions at state $s_{0,1}$
m	num. of alt. segments	n	num. of t. segments	a_α	action transit-to-thermal
a_β	action transit-to-target	k	decision epoch		

Figure 3.14: Illustration of the decision tree

left an updraft, the vehicle is not allowed to return to this particular updraft.

If r actions a_α are applicable from the initial state, $(r) \times (m \times n) + 1$ states are generated in the second layer. Note that *transit-to-target* actions are deterministic actions and thus feature only one outcome which is always a goal state. That is, goal states appear in all layers where $k > 0$. In other terms, a goal state can be reached from any of the non-goal states. Recall that this does not necessarily mean that the destination is indeed reachable. However, this formalization allows to generate local planning policies that are optimal with respect to the amount

of information that is currently at hand. At this point, the reader is also reminded about a key assumption in the strong definition of SSPMDPs, as provided by [76]. There exists at least one policy that will eventually terminate in a goal state, irrespectively of the initial state. With the goal state being applicable from any state, this key property of SSPMDPs is maintained.

Since the tree is constructed in a way that only *transit-to-target* actions can be applied from the second last layer ($k = N - 1$), all leaf states in layer N are goal states.

The pseudo code for the tree construction is stated in Algorithm 2.

The asymptotic time complexity of the tree construction can be stated as $O\left(|A||m \times n|^{N-1}\right)$ where the primitive operation is defined as the function which is called in line four of the algorithm.

Algorithm 2: Decision tree constructions - `constructdecisiontree()`

Input : MDP with ($\mathcal{A}$,s_0, m, n and horizon N)
Output: tree

```
1 for k = 0 to k = N − 1 do
2     for each s ∈ S_k do
3         for each a ∈ A do
4             calculate cost and probabilities of all m × n action outcomes
5         end
6     end
7     k = k+1
8 end
```

3.5.6. Planning Horizon Selection

For Vision-Based Autonomous Cross-Country Soaring (VBACCS), the planning horizon can be deduced from typical meteorological parameters and the vehicle's aerodynamic performance. Recall from Section 3.5.5, that the vehicle can execute at most N actions to reach the goal and the last action is always a *transit-to-target* action. By consequence, up to $N - 1$ updrafts can be approached.

Using vision for the generation of the map limits the perception horizon to the sensing range. Furthermore, updrafts feature a typical lifespan of $20\,\text{min}$ [23]. Thereby, the generated map is at most valid for a duration of $20\,\text{min}$. If we know the average time it takes to execute a *transit-to-thermal* action $\bar{t}_{a_\alpha}$, we can solve for the planning horizon according to

$$N = \frac{20\,\text{min}}{\bar{t}_{a_\alpha}} + 1. \tag{3.47}$$

As was shown in Equation (3.41), the duration of a transition to thermal action depends on the distance to the updraft, the climb rate and the vehicle's aerodynamic performance. Literature provides typical values for climb rates and the distance between updrafts [80]. Therefore, one can determine a representative value for $\bar{t}_{a_\alpha}$ and eventually solve for N.

3.6. Simulation Results

The primary purpose of this section is to illustrate the benefits of the proposed path planner by means of a simple scenario, assuming that an imperfect updraft map is provided. An additional analysis is conducted to study the required runtime of the suggested path planner on a typical state-of-the-art computer processor.

3.6.1. Numerical Example

In what follows, the 20 km cross-country soaring flight illustrated in Figure 3.15, defines a reference mission.

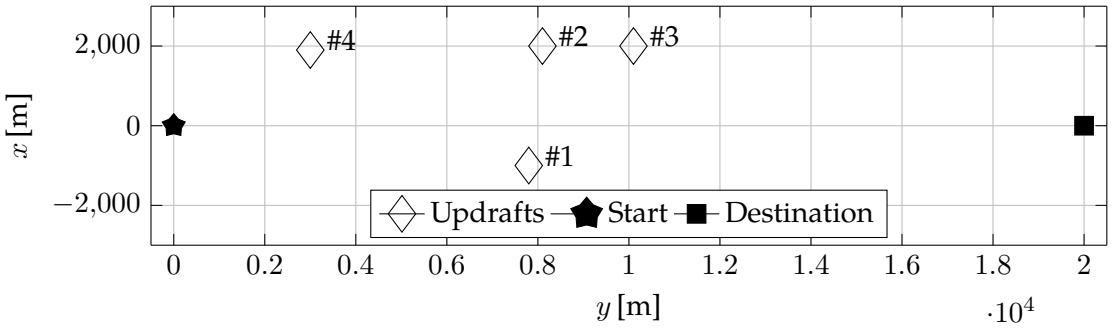


Figure 3.15: Reference scenario, xy-plane

A map with imperfect updraft parameters and related uncertainties is generated as input to the planning problem. Note that the uncertainties only refer to the vanish time and updraft climb rate, while the position is perfectly known.
Planning policies are calculated with and without considering map uncertainties. The map is selected such that the mean values allow to calculate a feasible solution to the destination. As a consequence, the deterministic planning policy will correspond to the policy that is obtained using the Floyd-Warshall-based path planning approach from [47]. This generates a comparison to the state of the art.
The simulations are performed using the simplified glider model from Section 3.2 with the polar curve parameters of the Cularis glider, as detailed in Appendix A. A one-shot planning is performed before the simulation and the map does not change while the vehicle is flying. Therefore, the validity of a calculated policy holds for the entire flight and it suffices to calculate the policy once.

Updraft Map and Meteorological Parameters Both the mean updraft parameters and the uncertainties are listed in Table 3.1 (map 1).

	Parameter	Unit	Updraft #1	#2	#3	#4
	$\bar{t}_v$	[s]	950	1000	1000	800
	$\bar{w}$	$[\frac{\text{m}}{\text{s}}]$	2	2	2	2
Map 1:	σ_{t_v}	[s]	200	150	100	10
	σ_w	$[\frac{\text{m}}{\text{s}}]$	0.15	0.15	0.15	0.15
Map 2:	σ_{t_v}	[s]	400	300	200	20
	σ_w	$[\frac{\text{m}}{\text{s}}]$	0.3	0.3	0.3	0.3

Table 3.1: Updraft map parameter settings

A second map with the same mean updraft parameters is generated. For the second map, increased uncertainties are selected by doubling the uncertainties of the first map (see map 2 in Table 3.1). This enables a parameter study w.r.t. varying map uncertainties. Here, uncertainties apply only to the vanish time and the climb rate while the updraft position uncertainties are supposed to have a negligible effect. The cloud base altitude for both maps is at z_{cb} =1200 m and the minimum allowable altitude is set to z_{min} =500 m defining an operational altitude range of 700 m.

For each of the two maps, 100 true world parameters are sampled from the normal distribution defined by the respective map parameters. Note that the true world parameters are sampled such that there exists at least one feasible solution to the target. Then, the simulation is run with the probabilistic planning solution and the deterministic planning solution one time per sample. This enables to collect statistical information w.r.t. the defined maps.

Path Planner Settings: The parameter k_1 is set to 0.01. The penalty for an outlanding, i.e. the parameter k_0 is selected to be 5000 s to significantly penalize outlandings in the decision-making process. The granularities for the state discretization parameters m and n are 5 and 3 respectively, and the confidence interval for the ellipses is chosen at 99 %.

Meteorological parameters are required to solve for the planning horizon N, as previously defined in Equation (3.47). More specifically, both the average spacing between updrafts and the climb rates are required for calculating the average time a transit to thermal action requires. Regarding the spacing between updrafts, different observations can be found in the literature. While both the authors of [81] and [82] measured updraft spacings of 2.5 km, the author of [83] observed updraft spacings of 10 km. For the calculation of the planning horizon, we refer to the smaller value which will also hold for the higher updraft spacing values

of 10 km. Typical updraft climb rates for cumulus soaring conditions are around $w = 2.5\,\mathrm{m/s}$ [80]. For the Cularis UAV, the MacCready airspeed corresponding to a climb rate of 2.5 m/s, i.e. $V_{MC}(w = 2.5\,\mathrm{m/s})$ is 16.1 m/s at a sink rate of v_{si} =1.53 m/s. Therefore, one can calculate a representative duration for an interthermal glide and climb sequence t_{a_α}

$$t_{a_\alpha} = t_g + t_c = \underbrace{\frac{2500}{16.1}\,[s]}_{t_g=155\,[s]} + \underbrace{\frac{1.53\,t_g}{2.5}\,[s]}_{t_c=95\,[s]} = 250\,[s].$$

Recall that updrafts last around 20 min[23]. This defines the maximum validity of the map $t_{max} = 1200\,\mathrm{s}$, and one can determine the planning horizon

$$N = \lceil \frac{1200}{250} \rceil + 1 = 6,$$

where $\lceil \cdot \rceil$ denotes the operator for rounding up to the next integer value.

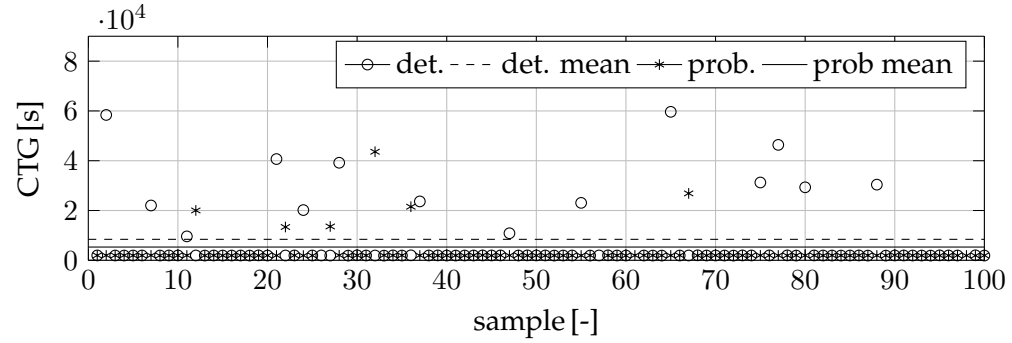
Evaluation: For evaluating the performance of both the deterministic and the probabilistic planning policies, both the mean realized Cost-To-Go (CTG) to the target, i.e. $\bar{\mathcal{C}}\,(s_1)$ and the corresponding standard deviations are calculated for the two uncertainty levels (i.e. for the two maps). Recall that the CTG is defined as the sum of the overall flight time and the additive cost term Ψ as introduced in Equation (3.46), in the case of an outlanding. The statistics in terms of both mean values and standard deviations over all 100 MC runs are listed in Table 3.2.

Map	Mode	Mean [s]	Std [s]
1	det.	8424	19652
	prob.	5321	46232
2	det.	33745	15626
	prob.	8260	17131

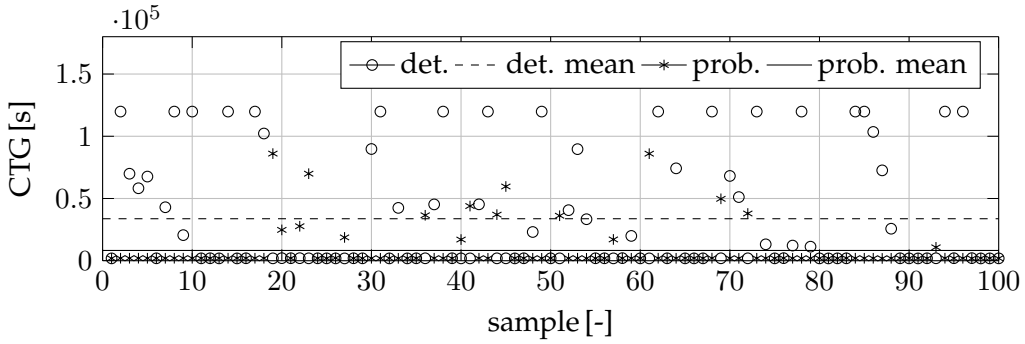
Table 3.2: Mean and standard deviation of the CTG

The mean cost of the probabilistic approach is approximately 1.5 times lower than its deterministic counterpart for the first map, i.e. at low uncertainties. At the increased uncertainty level, the probabilistic planning policy outperforms the deterministic approach by factor 4. Figure 3.16 illustrates the CTG per sample. For both cases, i.e. low and high map uncertainties, the CTG of the probabilistic planning solution varies less than for the deterministic results, demonstrating its robustness towards estimation errors. In both figures, the thin band with samples right above the x-axis correspond to successful flights, i.e. flights without outlandings and all samples lying above this band represent cases where the flight terminated in an

outlanding. There are samples where one of the solutions failed while the other succeeded and also some samples where both solutions failed or succeeded. Recall that the CTG only corresponds to the true flight time as long as the vehicle does not land out. That is, the CTG as illustrated in Figure 3.16 is not the average flight time since it also includes cases with outlandings. This explains why the maximum cost reaches up to approximately 60000 s.



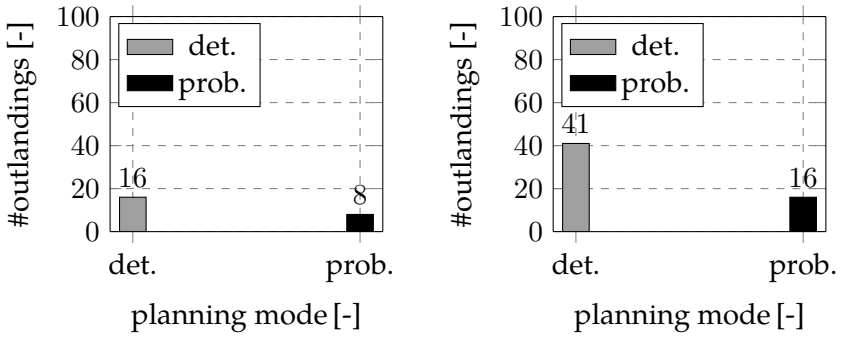
(a) Low map uncertainty (Map 1)



(b) High map uncertainty (Map 2)

Figure 3.16: Cost-To-Go (CTG) over sample

Another metric that illustrates the benefits of the probabilistic path planner is the cumulated number of outlandings (Figure 3.17). For both low (Figure 3.17a) and high (Figure 3.17b) map

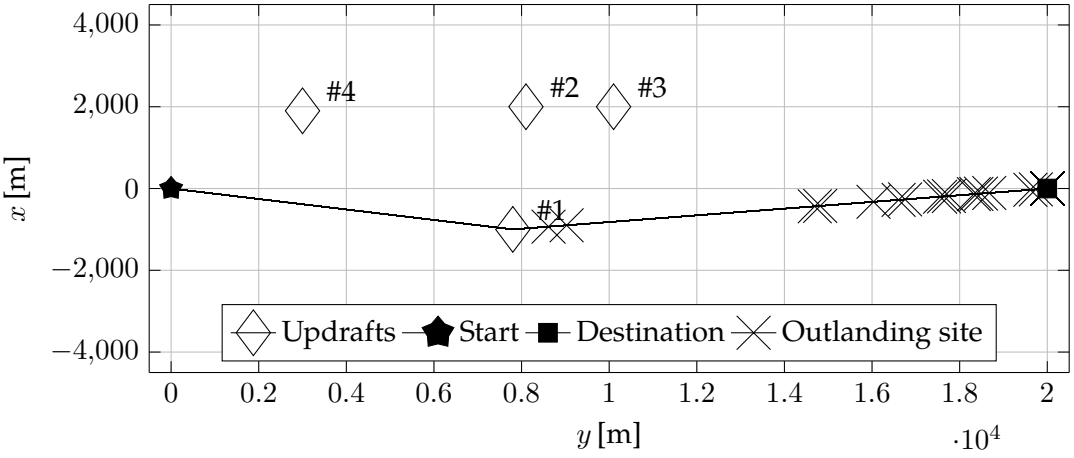


(a) Low map uncertainty (b) High map uncertainy

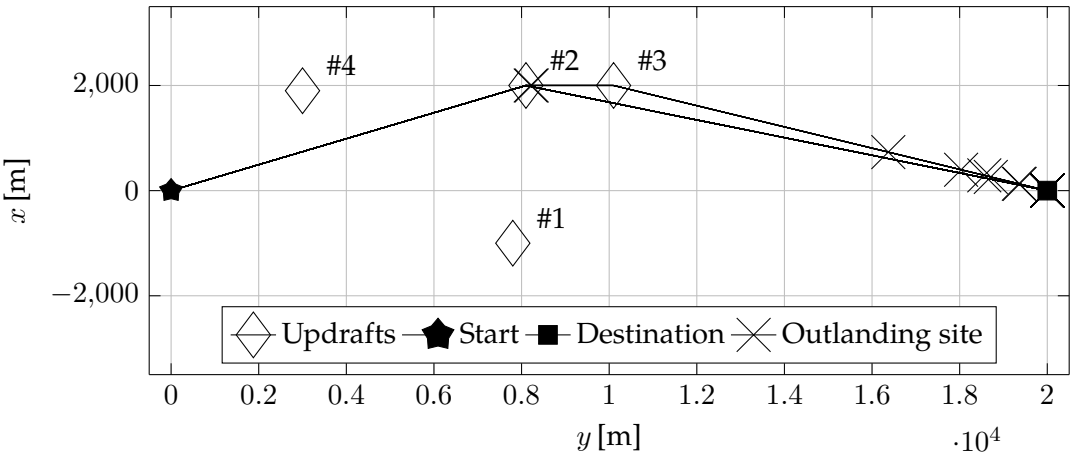
Figure 3.17: Cumulated number of outlandings

uncertainties, the number of outlandings is reduced from 16 to 8 and from 41 to 16, representing a reduction of 50% and 61% respectively, when relying on the suggested probabilistic path planner.

Figure 3.18 illustrates the vehicle's trajectories and outlanding sites (noted as a black x) resulting from the two planning policies for low map uncertainties.



(a) Deterministic



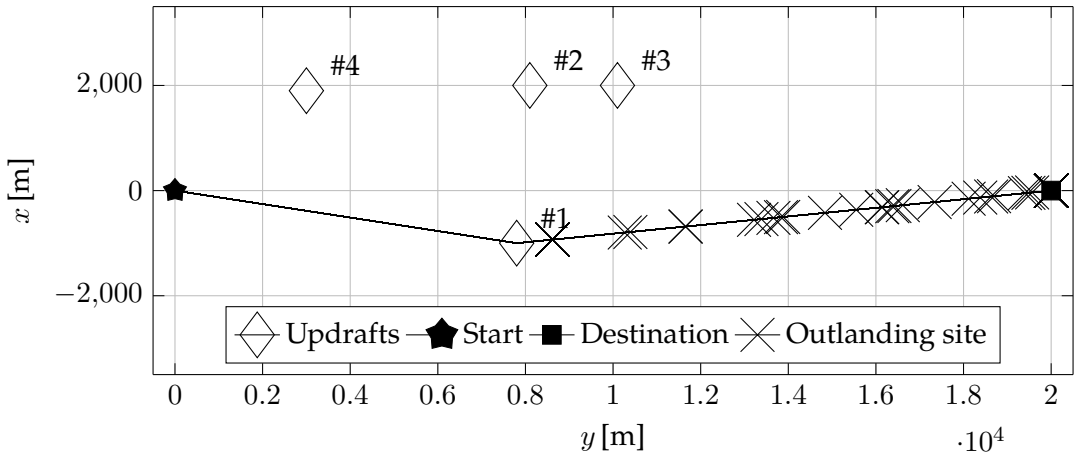
(b) Probabilistic

Figure 3.18: 2 D-Flight trajectories and outlanding sites at low map uncertainties

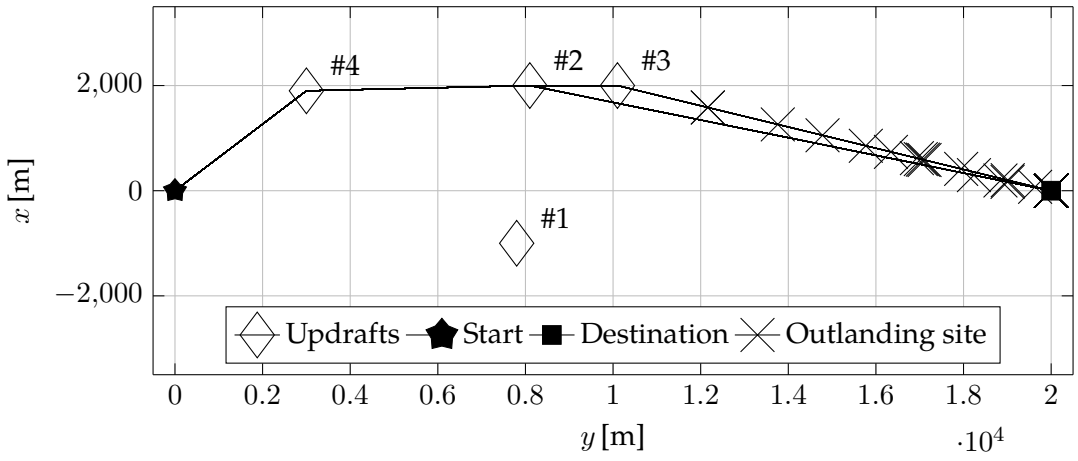
The trajectories in Figure 3.18a greatly reflect the shortcomings of the deterministic planning. The cost is optimized by selecting the seemingly fastest way. This is the trajectory from the origin over updraft #1 to the destination. Updraft #1 lies in proximity to the direct ground course. Thus, only a tiny detour has to be tolerated which minimizes the flight time when solely considering the mean of the estimated parameters. However, if the vanish time of updraft #1 does not allow to climb back to an altitude from which the destination can be reached without undercutting the minimum altitude z_{min}, the vehicle has no other option than to land out since the remaining updrafts do not lie in the vehicle's gliding range or have already vanished. Also

note that the deterministic planner only provides one planning solution. That is, no back-up plans are calculated since the map is assumed to be perfectly known. In comparison to the deterministic plan, the probabilistic plan suggests a more conservative trajectory, initially guiding the vehicle via cloud #2. This plan increases the additional flight time since approaching the updraft #2 involves a detour, compared to the approach over updraft #1. However, an overestimation of the related vanish time still leaves the chance open to regain altitude by using updraft #3 in case the destination is not reachable after terminating the climb in updraft #2. While the deterministic planner does not provide back-up plans, the probabilistic planning solution incorporates the necessitated back-ups for adapting the flight plan according to the true action outcomes.

A similar result is observed in Figure 3.19 where the flight trajectories are illustrated for the high map uncertainty case. Note that the trajectories of Figure 3.18a are identical to those in



(a) Deterministic



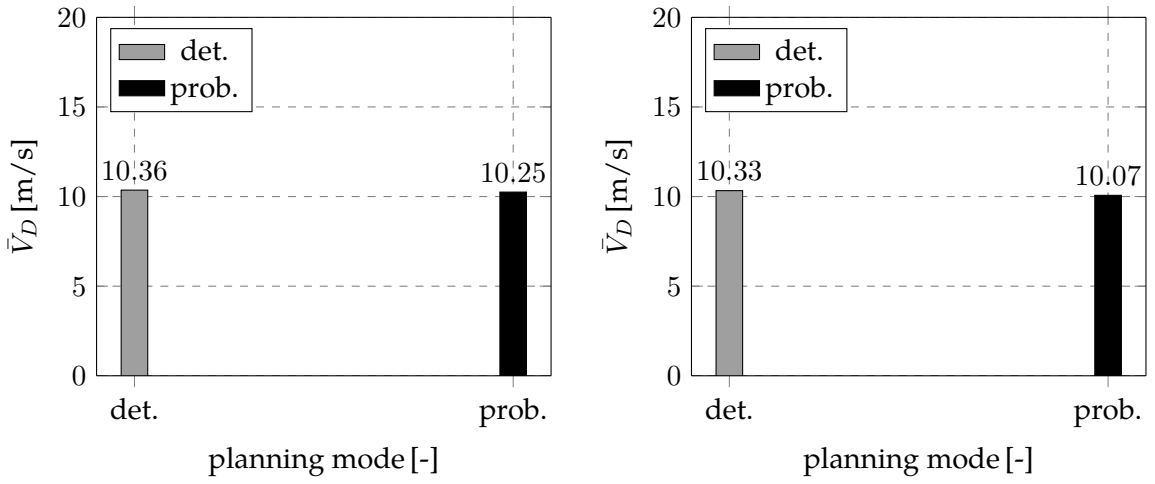
(b) Probabilistic

Figure 3.19: 2 D-Flight trajectories and outlanding sites at high map uncertainties

Figure 3.19a. This is because the deterministic policy is the same for the two uncertainty levels.

In comparison, the probabilistic planner suggests a plan which is adapted to the higher level of map uncertainties and initially guides the vehicle over updraft #4, still leaving the option of regaining altitude using updraft #2 and updraft #3 respectively.

A further metric to compare the two approaches is the mean speed to the destination $\bar{V}_D$ for successful missions, as illustrated in Figure 3.20. This parameter reflects the trade-off or cost that has to be tolerated in order to achieve less outlandings. As expected, the deterministic planning solution results in slightly higher ground speeds (approximately 1.1% and 2.5%). Note that the flight time increase induced by the probabilistic planner at high map uncertainties is due to the course deviation, when selecting updraft #4 as first action (see Figure 3.19). The small difference in the magnitude of the average speeds is however considered non-significant w.r.t. the reduction of outlandings.

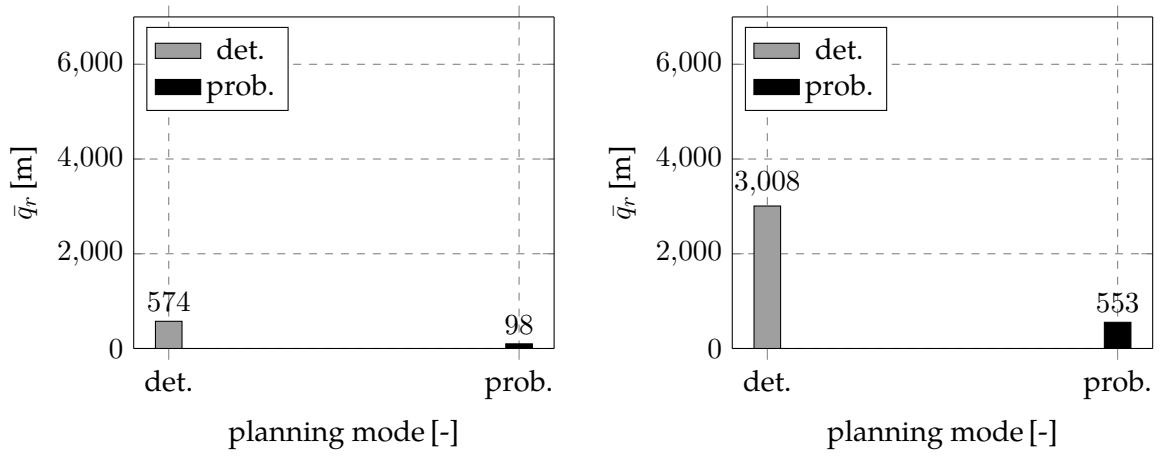


(a) Low map uncertainty

(b) High map uncertainy

Figure 3.20: Average speed to the destination

From a cross-country soaring point of view, the remaining distance to the destination constitutes a further evaluation metric. The average remaining distance for outlandings is illustrated in Figure 3.21. Though this parameter is included in the cost term and as such was already illustrated in Table 3.2 and Figure 3.16 respectively, it is purposely plotted here to demonstrate that the suggested path planner works as expected, i.e. outlanding sites of the probabilistic planner generally lie in the proximity of the destination compared to the deterministic planner. More specifically, the average remaining distance of the probabilistic path planner is approximately 80% less than for the deterministic path planner at both uncertainty levels.



(a) Low map uncertainty

(b) High map uncertainy

Figure 3.21: Average remaining distance

3.6.2. Computational Cost

The time complexity of the path planning algorithm consists of both the complexity of the tree construction algorithm and the complexity of the backwards induction algorithm. Thus, the time complexity Γ of the path planning is defined by

$$\Gamma = \underbrace{O\left(N|\mathcal{S}|^2|\mathcal{A}|\right)}_{\Gamma_{\text{bi}}} + \underbrace{O\left(|\mathcal{A}||m \times n|^{N-1}\right)}_{\Gamma_{\text{c}}}, \tag{3.48}$$

where the number of states is $|S| = \sum_{k=1}^{N-1} |S_{k-1}||m \times n||\mathcal{A}|^{k-1}$ and $|S_0| = 1$. Note that Equation (3.48) is a weak definition since the time complexity generally refers to primitive computing operations. While the "min" function, as called by the backwards induction (Algorithm 1) can be considered as a primitive operation, the "calculate cost and probabilities" function which is defined as the primitive function of the tree construction (Algorithm 2) requires more elementary operations, when solving the integrals for both the probability and the expected value calculation.

Another crucial property is the runtime. Though the runtime depends on the code efficiency and the machine on which the algorithm is executed, it is interesting to show typical values, especially when the target hardware which is designated to be carried on-board the UAV features a similar processor as the computer on which the simulations are conducted. For the tree construction, the runtime depends on the number of actions $|\mathcal{A}|$, the discretization settings m and n, as well as the planning horizon N. More specifically, the runtime depends on

the number of function calls for "calculate cost and probabilities", i.e. on the number of generated states. However, it is challenging to generalize the number of actions accessible from each node and thus to analytically quantify the average or maximum number of states which translate in an average or worst-case runtime. This is because the number of actions that can be executed from a state will generally decrease with the layer which sensitively impacts the cumulated number of states and so the runtime. To understand this action reduction, consider the tree construction starting from layer one onwards. At this particular time instance, all updrafts in the map are still active and a majority of updrafts can be assumed accessible. However, in the succeeding layers, time has elapsed and therefore the remaining lifespan of the updrafts has decreased which in turn reduces the number of applicable actions and thus the number of states.

In this work, we rely on the Monte Carlo method to identify both the average and the maximum (worst-case) number of states. This is done by generating a number of realistic inputs to the algorithm for capturing the statistics.

The number of states in the tree depends on the updraft density, the map size, both vanish times and updraft strenghts, and the state discretization parameters m and n respectively. Due to the circling trajectory of the UAV glider in thermal centering mode, the updraft map is best represented by a disk with a radius r_d around the vehicle's current position where r_d stands for the detection range of the vision sensor. Recall from Section 3.5.5, that course deviations up to $\pm 90^\circ$ w.r.t. the target are tolerated when approaching updrafts. Therefore, the map features an area of $0.5\pi r_d^2$ (see Figure 3.22).

Supposing both a perfect image processor and updraft estimator, the map contains all updrafts within the range $r < r_d$ which is a worst-case scenario for the computational load. One might argue that the map can contain more updrafts due to mismatched data association. However, the updraft position estimation is separated from the updraft strength and vanish time estimation, as was explained in Section 2.4.2. Further, the map only includes updrafts for which the vanish time and updraft strength estimation has been triggered. The trigger requires a consecutive series of matched vision measurements such that the threshold covariance $\gamma^\star$ is undercut. This procedure inherently reduces the likelihood of false updrafts in the map.

The detection range of the image processing sensor r_d is selected to be $5000\,\mathrm{m}$. The updraft density ρ_u can be defined as function of the updraft spacing d_u. For a regular distribution, we can rely on the density formula of a hexagonal grid. Thereby, the updraft density is defined

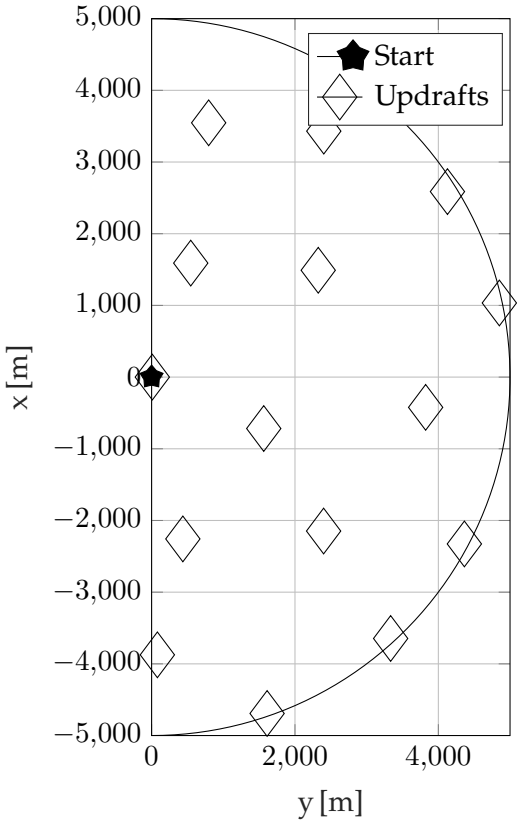


Figure 3.22: Half-disk-shaped updraft map

as

$$\rho_u = \frac{4}{\sqrt{3} d_u^2}. \tag{3.49}$$

By means of the updraft density, we can calculate the number of updrafts that lie in a given area of size A

$$n_u = \rho_u A. \tag{3.50}$$

This yields $n_u = 0.5\rho_u \, \pi r_d^2$. updrafts for the half disk. The parameters used to generate sample trees are listed in Table 3.3.

Parameter	Value	Unit
d_u	2.5	[km]
r_d	5	[km]
n_u	15	[-]
t_v	$\mathcal{U}(0, 20)$	[min]
w	$\mathcal{N}(2.5, 0.5)$	[m/s]
m	5	[-]
n	3	[-]
N	6	[-]

Table 3.3: Monte Carlo settings for generating sample trees

In Figure 3.23, both the average number of states per layer (average MC), and the worst-case number of states per layer (worst-case MC), as obtained from a 100-run Monte Carlo study are illustrated in a semilogarithmic plot.

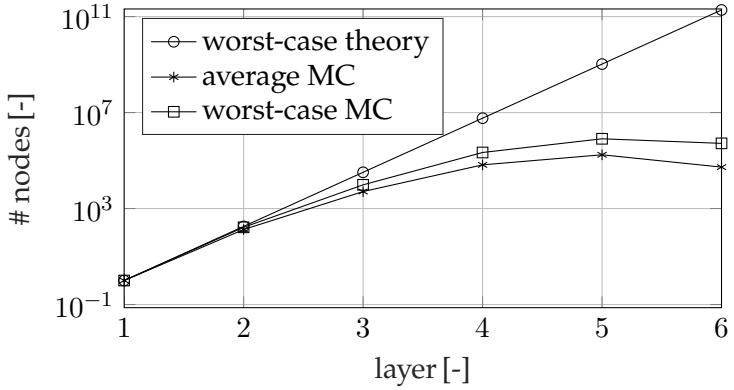


Figure 3.23: Nodes per layer

In the same figure, the theoretical worst-case number of states per layer for $m = 5$ and $n = 3$ is likewise shown. Note that this worst case holds under the condition that all updrafts are accessible from every state. As expected, both the maximum and average number of states at each layer, resulting from the Monte Carlo study lie significantly lower than for the theoretical worst-case. Also note that for increasing decision epochs, the number of states decreases. This is due to the fact that only a fraction of the updrafts last long enough to enable "long" routes, i.e. routes where the vehicle visits more than 4 updrafts before transitioning to the target. To illustrate this effect independently of the discretization parameters m and n, the ratio between the number of states obtained from the MC study and the theoretical worst-case is illustrated in Figure 3.24. The reduction of states becomes especially significant in the third

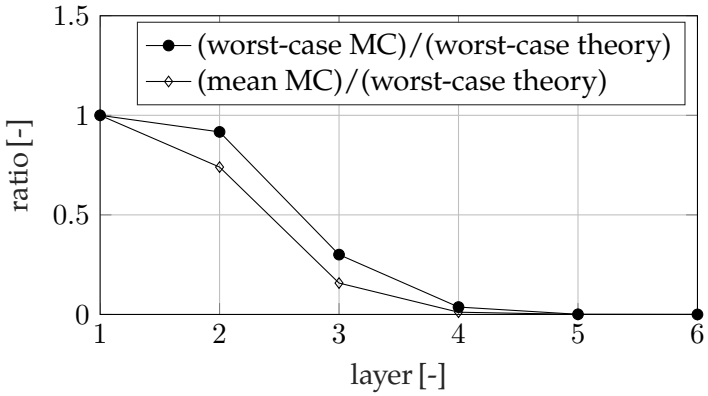


Figure 3.24: State number ratio between the theoretical worst-case scenario and the Monte Carlo results

and fourth layer, where the Monte Carlo study predicts only 25% and 3% of the theoretical

worst-case respectively. The corresponding runtimes for both the tree construction and the value iteration are listed in Table 3.4. Note that the algorithms were programmed in Matlab®, mex-compiled, and executed on an Intel Core i7 processor. Also note that miniaturized COTS computing systems with equivalent specifications and dimensions compatible with small UAV gliders are readily on the market. As expected, the runtime required for the tree construction is significantly higher than the runtime for the backwards induction. More specifically, it requires about 16 times more to calculate the tree than to solve the MDP. However, for realistic input settings (number of updrafts) and a worst-case scenario where all updrafts are identified by the estimator, an average runtime of approximately 6 s is still considered to be fast enough regarding the fact that the relative motion between the vehicle and the observed clouds is slow - resulting in slow changes of the updraft maps which does not require for high re-planning rates. Additionally, the code was not optimized w.r.t. runtime which means the runtime can still be reduced, for example through vectorization.

		TC Runtime [s]	BI Runtime [s]	Cumulated Runtime [s]
	mean	5.2674	0.3561	5.6235
MC	std	6.9427	0.0122	6.9548
	wc	25.4021	0.3956	25.7894

Table 3.4: Runtime statistics

3.7. Summary

A Markov Decision Process (MDP)-based path planner for Map-Based Cross-Country Soaring (MBCCS) is suggested in this chapter. The path planner takes updraft maps with uncertainties in the updraft vanish times and updraft strengths as input and calculates planning policies that balance between flight time and the risk of outlandings. A simple flight dynamics model is introduced for propagating the state of the glider. Furthermore, a final glide law is proposed, for calculating airspeed commands during final glide. The technique of Unscented Transform (UT) is utilized in order to transform continuous map uncertainties into uncertain states. The discretization of the outcome probabilities and expected values is done using analytical integration rather than iterative numerical integration in order to reduce the computational load in the decision tree construction. The cost function for the planning problem is based on a glider competition-like rule with penalties for outlandings. The performance of the proposed probabilistic path planner is compared to a state-of-the-art deterministic planner, demonstrating its potential to significantly improve robustness towards updraft map uncertainties thus reducing the risk of outlandings and providing a powerful planning solution for Vision-Based Autonomous Cross-Country Soaring (VBACCS) where updraft maps are inherently uncertain. Additionally, a study is conducted, demonstrating that the proposed probabilistic planner is shown to be implementable on a small computer, enabling an on-board execution using Commercials Off-The-Shelf (COTS) electronic devices.

Chapter 4

Simulation Evaluation

As pointed out earlier, the main concern of this thesis is to contribute towards more reliable autonomous soaring. Therefore, in Chapters 2 and 3, methods for remote updraft estimation and map-based path planning algorithms are proposed. In the corresponding chapters, both of these contributions are assessed individually. In this chapter, a joint evaluation of the two components is presented. The evaluation is conducted against a variation of crucial parameters. Additionally, the system's performance both in terms of flight time and outlandings is compared to the performance of the state-of-the-art cross-country soaring approach, from the authors of [3].

The estimation and path planning methods presented in this work are platform-independent. While a simplified aircraft model was utilized in Chapter 3, a nonlinear 6 DOF aircraft model (Cularis electric glider model, see Figure 4.1 , specifications noted in Appendix A) in combination with a set of low-level flight controllers is used in this chapter to generate realistic UAV glider flight dynamics. For obtaining statistically representative performance measures of the proposed methods, a series of Monte Carlo (MC) simulations is conducted.

Figure 4.1: Cularis electric glider (image taken from `http://multiplex-rc.de/`)

4.1. System Architectures

In order to highlight the benefits of both the remote updraft estimation and the probabilistic path planner, the simulations in this chapter are repeated for different system architectures. That is, we define three systems that consist of different constellations of estimation and planning components. The first one aims to reproduce the state-of-the-art performance in autonomous cross-country soaring. Therefore, the glider flies without any remote updraft sensing which reproduces flight performances that are obtained using the techniques of [3], i.e. only in-updraft estimation. The two other architectures rely on the vision-based remote updraft estimation from Chapter 2 where the second system architecture is based on the deterministic path planner and the third one on the probabilistic path planner. This allows for both a rigorous and sequential evaluation of the elaborated navigation and planning methods.

Thermal Centering Control For all three system architectures, in-updraft identification is realized by monitoring the vehicle's energy state. Note that the energy state is defined by both the first and the second derivative of the total energy ($E = mgh + 1/2mV_a^2$), i.e. both total energy rate $\dot{E}$ and acceleration $\ddot{E}$. A mode logic, suggested by the authors of [2], outputs an "in-updraft flag" as soon as the vehicle's energy state passes a predefined threshold.
Further, all three architectures rely on the same thermal centering control law which is inspired by the work presented in [20]. The centering control law builds on three rules of thumb in order to generate bank angle controls that are function of the vehicle's energy state

1. As climb improves, decrease the bank angle,
2. As climb deteriorates, increase the bank angle,
3. If climb remains constant, maintain a constant bank angle.

These three rules are implemented to generate a commanded heading derivative

$$\dot{\Psi}_{cmd} = \dot{\Psi}_{ss} + K_{tc}\ddot{E}, \tag{4.1}$$

where $\dot{\Psi}_{ss}$ is a steady state feed forward contribution which is function of the airspeed and some selected steady state radius r_{ss} such that

$$\dot{\Psi}_{ss} = V_a/r_{ss} \tag{4.2}$$

and K_{tc} is the control law gain. Note that the turn rate Equation (4.2) is an approximation, holding for small flight path angles which is mostly the case for glider planes. The bank angle command for thermal centering is then defined assuming a coordinated turn by

$$\Phi_{cmd} = \arctan\left(\frac{V_a}{g}\dot{\Psi}_{cmd}\right). \tag{4.3}$$

System Architectures The three system architectures are detailed as follows:

1. **Blind:** The first architecture is non-vision based, i.e "blind". Updrafts are thus not remotely sensed and the vehicle "blindly" flies towards the target, switching to thermal centering mode whenever the presence of an updraft is identified, given the vehicle's energy state. For this architecture, airspeed is always set to the value of the best glide angle. Thermal centering mode is maintained until the glider reaches the cloud base altitude or the updraft has vanished. The functionality of the "blind" architecure is explained in the flowchart Figure 4.2.

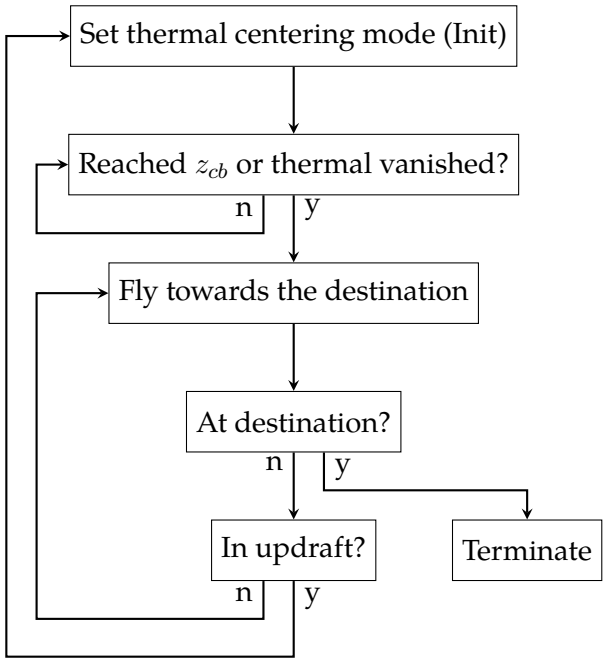


Figure 4.2: Flowchart of the blind system architecture (Without remote updraft estimation)

2. **Deterministic:** The second system architecture (Figure 4.3) relies on the remote updraft estimation as introduced in Chapter 2 for perceiving updrafts in the vehicle's environment. The path to the target is computed using the deterministic path planning approach from Chapter 3. The path planning is executed at a fixed sample rate - thus constantly adapting the planning solution to the updated updraft map. The approach to an updraft

is separated into two consecutive segments. In the first segment, the glider flies towards the updraft with the MacCready airspeed setting. Second, when the glider is in proximity of the estimated updraft position, airspeed is reduced from the MacCready airspeed to the minimum sink airspeed, and the energy state of the vehicle is monitored for a period of maximum 60 s. If an updraft is identified within this time window, the vehicle is set to thermal centering mode. If no updraft is identified, the updraft is deleted from the map and the next position is approached.

3. **Probabilistic:** The third system architecture has the same functionality as the second architecture but relies on the probabilistic path planner instead of the deterministic one.

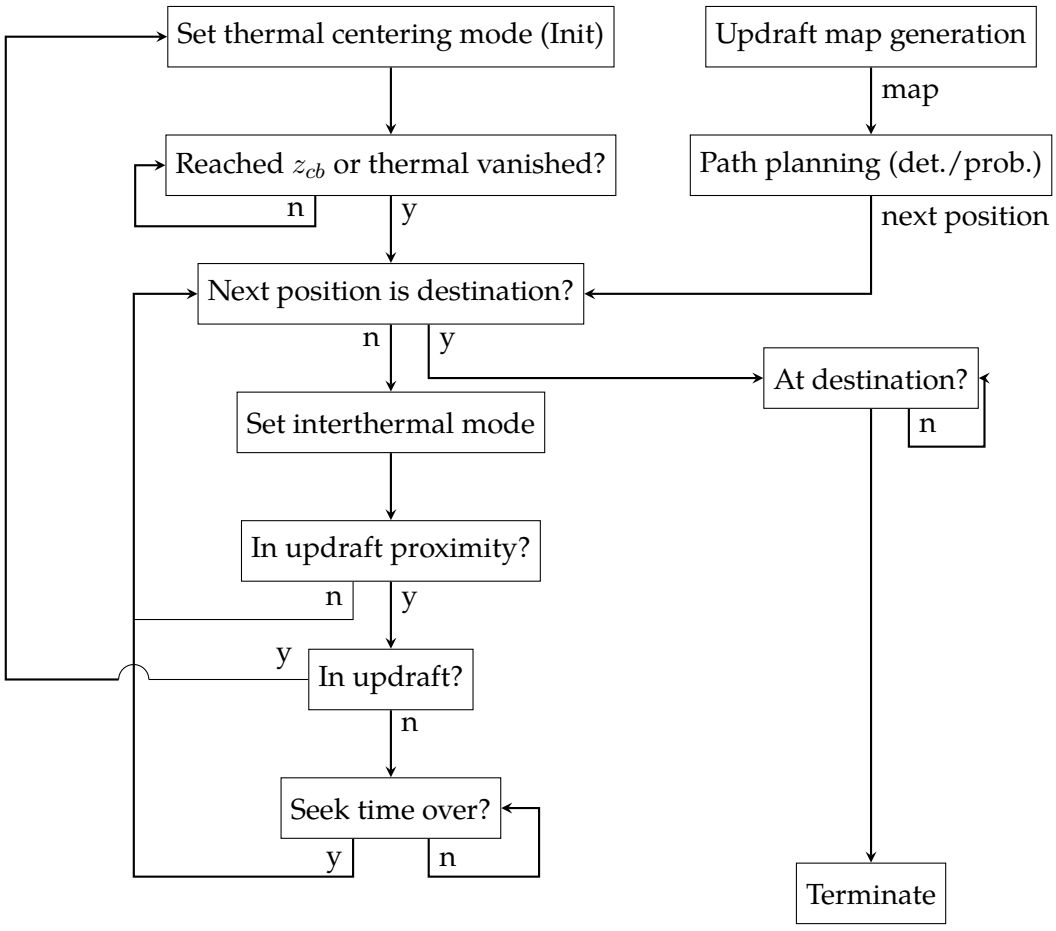


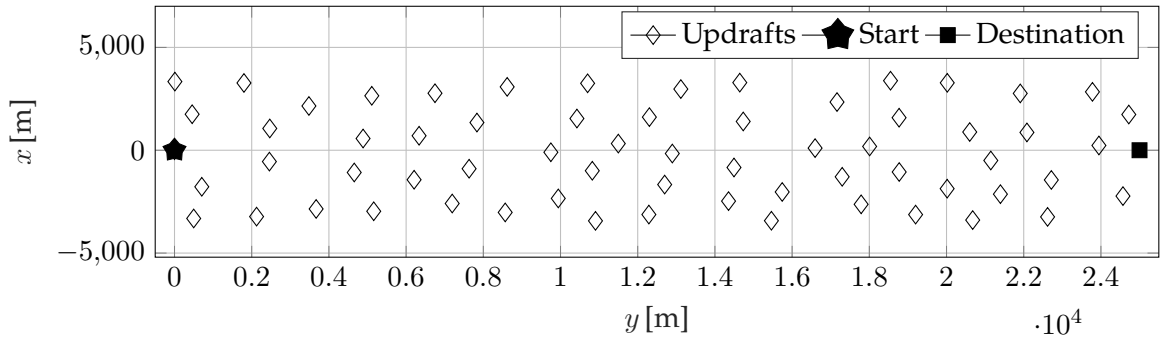
Figure 4.3: Flowchart of the deterministic and the probabilisitc architecture (With remote updraft estimation)

4.2. Monte Carlo Simulations

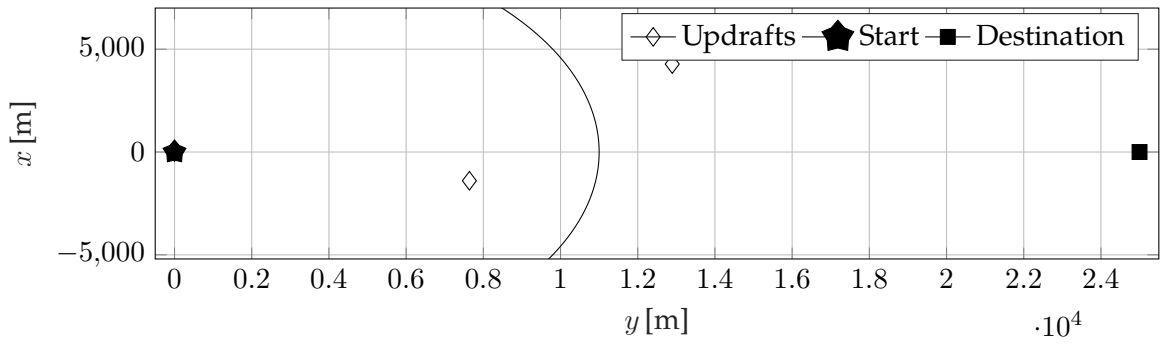
For studying the performances of the three system architectures, a series of Monte Carlo simulations is conducted.

4.2.1. Mission

The benchmark mission of the Monte Carlo Simulation (MCS) is a 25 km cross-country soaring flight where updrafts are randomly located within a 25 km $\times$ 7 km weather tile around the direct line between the start point and the target (Figure 4.4). The vehicle is released at the origin at an altitude of 1100 m in thermal centering mode and its task is to fly to the destination. Recall that the vehicle is not allowed to undercut a minimum altitude, selected to be $500\,m$. Thus, mission success is whenever the vehicle reaches the destination. The simulation is stopped as soon as an outlanding occurs, i.e. the vehicle's altitude undercuts the minimum altitude or whenever the vehicle reaches the target.



(a) High updraft density



(b) Low updraft density

Figure 4.4: Example scenarios with high and low updraft density

4.2.2. Monte Carlo Setups

In cross-country soaring, the spacing between updrafts has a major impact on the mission success. Since this parameter can significantly vary, we define two different levels of updraft densities to increase the validity of the simulation results and also to investigate the robustness of the individual architectures. A further crucial parameter for the two vision-based system architectures is the image processing noise, as introduced in Equation (2.25). Therefore, as for the updraft density, two levels of image processing noise are defined. This generates four MC setups, i.e. the permutations of low/high updraft density and low/high image processing noise. For each of these four setups, a total of 100 MC simulation runs is conducted for both the second and the third system architecture. Since the performance of the "blind" system architecture is independent from the image processing noise, MC simulations for this architecture are solely conducted at the low and the high updraft density settings. Note that the scenarios are feasible, i.e. there exists at least one solution to the target.

Both the updraft spacings d_u and the image processing noise levels $\Sigma_{\mathbf{y}_{LR}}$ defining the four MC setups are listed in Table 4.1. The values for the updraft spacings are taken from the literature, where the lower spacing refers to the findings of [81, 82], and the upper updraft spacing to ones of [83]. Regarding the setting of the image processing noise, the default level from Chapter 2 is selected as low level, and twice its magnitude is selected to define deteriorated conditions.

Parameter	Values		Unit
	low	high	
$\Sigma_{\mathbf{y}_{LR}}$	$2\,\mathbf{I}_{4\times4}$	$4\,\mathbf{I}_{4\times4}$	[px]
d_u	2.5	10	[km]

Table 4.1: MC setup parameter values

For each MC simulation run, n_u updrafts with random position, updraft strength, and lifespan are generated. Recalling the updraft density formula from Equations (3.49) and (3.50), n_u is calculated according to

$$n_u = \rho_u A_T = \frac{4}{\sqrt{3} d_u^2} A_T, \tag{4.4}$$

where A_T represents the size of the weather tile. For the given tile size of 25 km $\times$ 7 km, we obtain 65 and 4 updrafts at the defined spacings of 2.5 km and 10 km. During the simulation, vanishing updrafts are replaced by new updrafts, again with random position, lifespan and strength, thus maintaining a constant updraft density over the course of the simulation. Figure 4.4 illustrates the mission setup with varying updraft densities.

Table 4.2 specifies typical meteorological parameters that can be encountered during the summer in the northern hemisphere. The values for both updraft lifespans and strengths, as well as the cloud-base altitude, and the convexing mixing layer thickness z_i are taken from [80, 23]. The average cloud width amplitude $\bar{d}_{max}$ is set to $750\,\mathrm{m}$ which according to Equation (2.19) converts into a factor κ of $0.0033\,1/\mathrm{s}$. The perturbation gain from Equation (2.6) for generating variating updraft strengths is selected to have a standard deviation of 0.12.

Parameter	Distribution	Unit
z_{cb}	const.,1200	[m]
z_i	const., 2000	[m]
κ	0.0033	[1/s]
c	$\mathcal{N} \sim (1, 0.12)$	[-]
$w^\star$	const., 3	[m/s]
τ_u	$\mathcal{N} \sim (20, 2)$	[min]

Table 4.2: Setup-independent meteorological parameter values

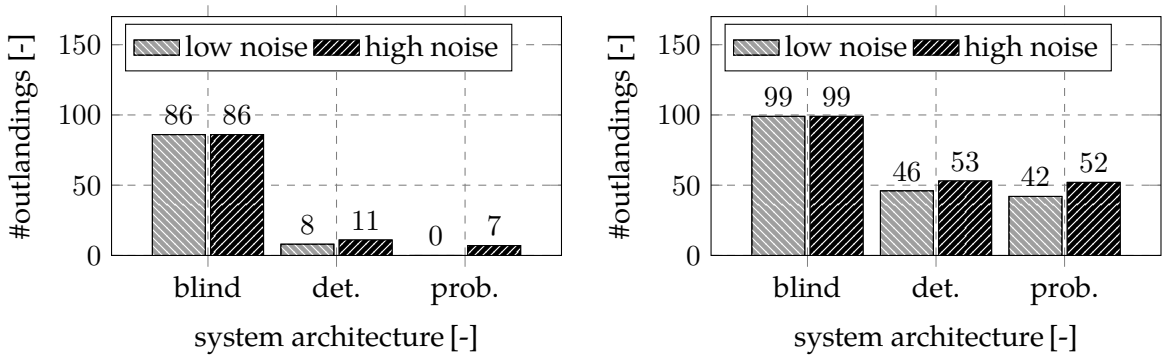
The major parameter settings for both the estimator and the path planner are listed in Table 4.3. Note that the sample time of the remote updraft estimation as well as the path planning parameters are the same as used in Chapters 2 and 3. Also note that the two parameters k_0 and k_1 from Equation (3.42) are selected to be conservative, to prioritize avoiding outlandings over a reduction of flight times.

Parameter	Description	Value	Unit
$t_{s,est}$	Remote estimation sampling time	0.1	[s]
$t_{s,pp}$	Path planning sampling time	30	[s]
k_0	Penalty for outlanding (constant)	5000	[s]
k_1	Penalty for outlanding (remaining distance)	0.01	[-]
m	Altitude discretization	5	[-]
n	Time discretization	3	[-]

Table 4.3: Major path planning and estimation parameter value settings

4.2.3. Outlanding Statistics

Since the main purpose of the vision-based soaring approach is an augmentation of the reliability of autonomous cross-country soaring systems, the cumulated number of outlandings constitutes the most important figure of merit. Figure 4.5 illustrates the cumulated number of outlandings over all Monte Carlo (MC) runs as function of the different image processing noise and updraft density levels. As illustrated in Figure 4.5a, at high updraft density, the "blind"



(a) High updraft density

(b) Low updraft density

Figure 4.5: Number of outlandings depending on image processing noise and updraft density

approach yields a total number of 86 outlandings over 100 runs, clearly indicating the limits of autonomous cross-country soaring without remote-updraft estimation. In comparison, the map-based deterministic planning results in only 8 outlandings, i.e. a significant reduction of 78 outlandings is obtained, thanks to both the remote updraft estimation and the deterministic planner. As expected, when relying on the probabilistic path planner, the overall number of outlandings can further be reduced. More specifically, an additional reduction of 100% of the number of outlandings achieved with the deterministic planner is obtained. This demonstrates the benefits of the probabilistic planner in terms of outlandings in comparison to its deterministic counterpart.

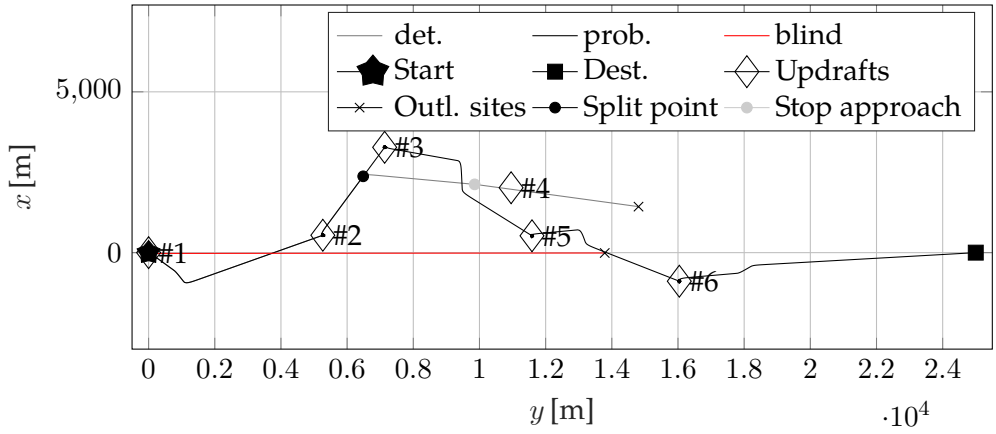
In the same figure, the number of outlandings at high image processing noise are listed. Since the blind system architecture is insensitive to image processing noise, it results again in 86 outlandings over all runs. Evidently, the increased image processing noise causes an increased number of outlandings for the two vision-based system architectures due to a degradation of the remote updraft estimation. More precisely, the number of outlandings for the deterministic planner increases by 3 while the outlandings caused by the probabilistic planner increase by 7. As for the low measurement noise, the trend remains, i.e. the probabilistic architecture causes again less outlandings, compared to the deterministic planner.

Figure 4.5b depicts the number of outlandings for low updraft density at both image processing noise levels. Interestingly, the "blind" system architecture not only degrades, but features a mission fail rate of 99%. At low updraft density and low image processing noise, the vision-based approaches have success rates of 54% for the deterministic and 58% for the probabilistic planner, i.e. the probabilistic planner still outperforms the deterministic one. For low updraft density and high measurement noise the two vision based approaches feature comparable performances. That is, the probabilistic planner yields one outlanding less than its deterministic counterpart. Figure 4.5 further reflects how the updraft spacing/density impacts the success rate. While the "blind" performance features poor success rates at both density levels, and as such is relatively insensitive to the regarded updraft density, the vision based approaches remain sensitive to both the updraft density and the image processing noise with a success rate variatin from 47% to 92% for the deterministic, and from 48% to 100% for the probabilistic path planner respectively.

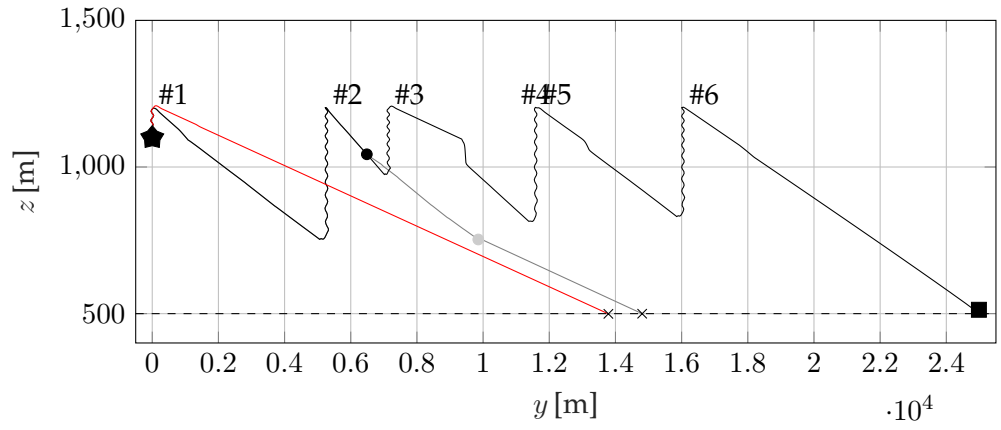
4.2.4. Analysis and Comparison of Flight Trajectories

In this section, a representative MC simulation sample is studied to illustrate the different functionalities of the three system architectures. Using the same meteorological setup with an updraft spacing of 2.5 km, and the same noise seed, the flight is repeated three times, each time with another system architecture.

Figure 4.6 illustrates the three flight trajectories as well as the relevant updrafts. The red solid



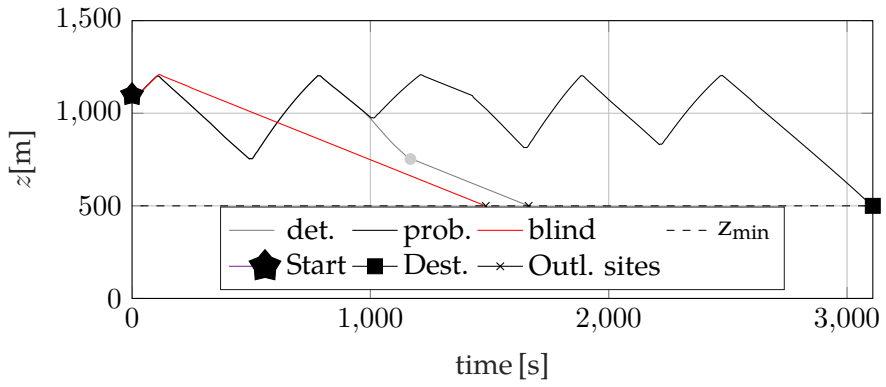
(a) xy-plane



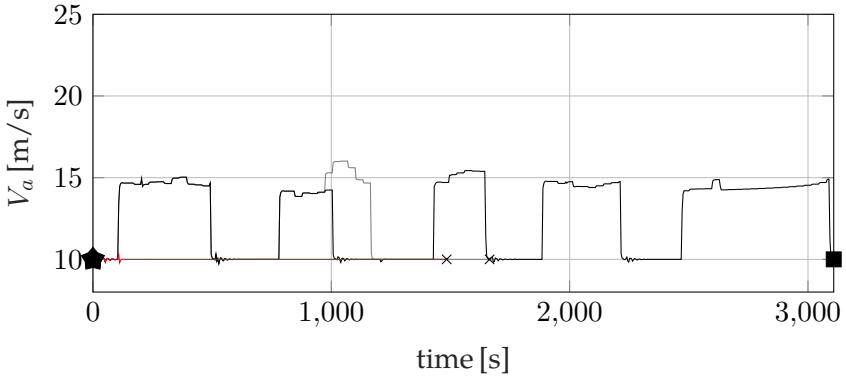
(b) yz-plane

Figure 4.6: Flight trajectory comparison (position)

line results from the "blind" architecture, and the gray and black solid lines from the deterministic and probabilistic architectures. We will first study how the remote updraft estimation impacts the vehicle's altitude and airspeed profile. Therefore, consider Figures 4.6b and 4.7, where both the altitude and airspeed profiles are illustrated. Without remote updraft estimation, the glider's only option in minimizing the distance to the target lies in a direct approach to the target, flying at the airspeed of the best glide ratio, as illustrated in Figure 4.7b. Only updrafts that lie on the direct path to the target can be utilized. With no updrafts lying on its



(a) Altitude profiles



(b) Airspeed profiles

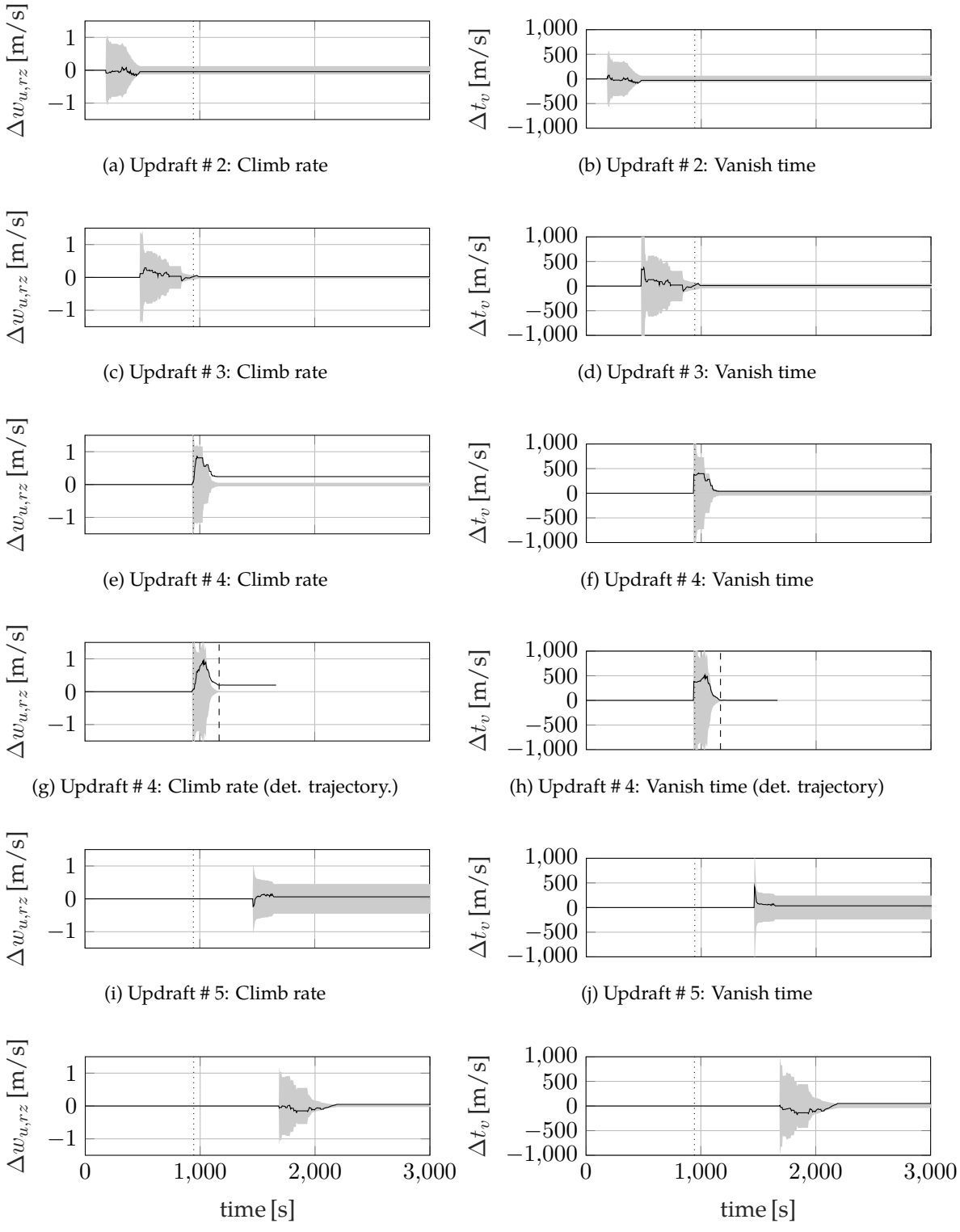
Figure 4.7: Flight trajectory comparison (altitude and airspeed)

path, the flight terminates at approximately 1500 s without reaching the target. Note that the outlanding sites are indicated with an x.

Conversely, by means of the remote updraft estimation in combination with the map-based path planning, up to 5 additional updrafts (updraft #2 - #6) can be visited for regaining altitude. As shown in Figure 4.7b, the remote updraft estimation enables the glider to cross distances at the MacCready airspeed settings which are about 50% higher than the velocity of minimum sink. Note that the segments where the airspeed setting is about 10 m/s correspond to thermal centering flight, i.e. the vehicle's altitude increases during these periods which can be seen in the corresponding altitude profile (Figure 4.7a).

It is also interesting to compare the flight trajectories of the deterministic and the probabilistic system architecture. These trajectories coincide until a split point that lies between the updrafts #2 and #3 (Figure 4.6). At the split point, the probabilistic planner continues to guide the vehicle towards updraft #3 and the deterministic planner suggests to approach the newly identified updraft #4. However, the updraft #4 vanishes sooner than estimated which is why the vehicle cannot benefit from it. As a consequence of this particular decision, the vehicle

under the deterministic policy has to land out. To illustrate the origin of the different decision-making at the split point, the estimation errors of the updrafts #2 - #6, observed during the flight with the probabilistic planner are illustrated in Figure 4.8.



(a) Updraft # 2: Climb rate

(b) Updraft # 2: Vanish time

(c) Updraft # 3: Climb rate

(d) Updraft # 3: Vanish time

(e) Updraft # 4: Climb rate

(f) Updraft # 4: Vanish time

(g) Updraft # 4: Climb rate (det. trajectory.)

(h) Updraft # 4: Vanish time (det. trajectory)

(i) Updraft # 5: Climb rate

(j) Updraft # 5: Vanish time

(k) Updraft # 6: Climb rate

(l) Updraft # 6: Vanish time

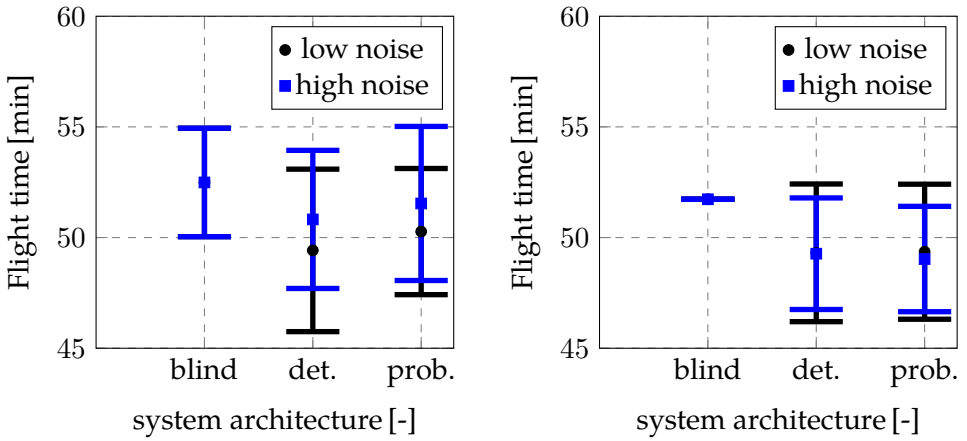
Figure 4.8: Estimated updraft parameters

The parameters of updraft #4, as observed using the deterministic architecture (Figures 4.8g and 4.8h) are additionally illustrated fore the sake of explanation. The gray shaded areas indicate the 3-σ error intervals of the parameters and the dotted vertical lines in all figures indicate the split time. Notice that the updraft estimates are triggered at different times. This is because the vehicle successively perceives the updrafts during the flight. Furthermore, we can see significant variations in the quality of the parameter estimates. These variations result from different observability conditions. Individual clouds are only visible for short periods of time, compared to the entire mission length. This explains why the estimates freeze when no more measurements are available.

The trajectories of the probabilistic and the deterministic architecture coincide up to the split point. Therefore, both planners have exactly the same state estimates at their disposal until this point. However, both the estimates of the updraft strength and vanish time of the fourth updraft (see Figures 4.8e to 4.8h) are of poor quality. More specifically, the vanish time is overestimated by approximately 400 s. This overestimation and the target-proximity of the updraft explain the choice of the deterministic planner. While approaching the updraft, the estimate converges until the point where the path planner judges that an approach is not beneficial. This instance is illustrated by a gray point (stopp approach point) in Figures 4.6, 4.6b and 4.7a and also in Figures 4.8g and 4.8h, by the dashed vertical line. The vehicle has no other option than to land out since a potential approach to any other updraft would violate the minimum altitude constraint when flying with the MacCready speed. Note that this outlanding could have been avoided by a reduction of the MacCready airspeed. Thereby, this example also reveals a potential refinement of the proposed path planner that can be obtained by optimizing both airspeed and updraft selection in parallel. Contrary to the deterministic planner, the probabilistic one exploits the uncertainty information of the updraft estimates and continues to guide the glider towards updraft #3 which bears only a neglectable risk of a missjudged approach. Figures 4.8g and 4.8h illustrate that the uncertainties related to the parameters of updraft #3 are of small magnitude which enables a more precise prediction of the action outcome. As a consequence of anticipating the risk in the decision-making, the vehicle accomplishes the mission without an outlanding.

4.2.5. Flight Time Statistics

As a second figure of merit, the flight time for successful flights is studied. The flight time statistics for all four MC setups are illustrated and denoted in Figure 4.9 and Table 4.4 respectively.



(a) High density (b) Low density

Figure 4.9: Flight time statistics

	System Architecture			Unit
	blind	det.	prob.	
mean	52.49	49.42	50.27	[s]
std	2.45	3.67	2.85	[s]

(a) Low noise, high density

	System Architecture			Unit
	blind	det.	prob.	
mean	52.49	50.81	51.54	[s]
std	2.45	3.41	3.48	[s]

(b) High noise, high density

	System Architecture			Unit
	blind	det.	prob.	
mean	51.73	49.31	49.36	[s]
std	0	3.11	3.05	[s]

(c) Low noise, low density

	System Architecture			Unit
	blind	det.	prob.	
mean	51.73	49.28	49.03	[s]
std	0	2.52	2.38	[s]

(d) High noise, low density

Table 4.4: Flight time statistics

The flight time statistics at high updraft density are illustrated in Figure 4.9a and Tables 4.4a and 4.4b. Note that the blind system architecture is insensitive to image processing noise and thus features equal performances at both noise levels. Independently from the image procesing noise, at high updraft density, the deterministic architecture yields the lowest mean flight time to the target with 49.42 and 50.81 min respectively. One could expect the blind system architecture to yield the lowest mean flight time because no detours are flown. This is however not the

case since the vehicle does not fly with the MacCready airspeed but with the airspeed of the best glide ratio which is considerably lower. At both image processing noise levels, the probabilistic architecture results in a mean flight time increase of approximately 1 min, compared to the deterministic architecture thereby reflecting the more conservative planning approach. The flight time increase of around 1.5% can however be tolerated regarding the significantly higher success rate of the probabilistic architecture. Notice that at increased measurement noise, the mean flight time of both vision-based approaches increases. This is because the degradation of the estimation causes more missed approaches to updrafts. That is, if the vanish time of an updraft is highly overestimated, the corresponding flight leg adds to the flight time but the vehicle does not gain energy. The standard deviations of the flight times for both vision-based architectures are slighty higher than for the blind system architecture. This effect is mainly introduced by the randomness of the updraft positions that can create significant flight detours.
Figure 4.9b and Tables 4.4c and 4.4d show the flight time statistics at low updraft density. The blind architecture results in a flight time of 51.73 min for the one successful flight. At both noise levels, the vision-based approaches result in comparable mean flight times. This is because the decision-making process is less complex than for the high updraft density case, increasing the likelihood of both planners to make similar decisions. Interestingly enough, the flight times at low updraft density are lower than at high updraft density. This is explained by the fact that less updrafts are visited. Therefore, less time is spent in thermal centering mode which evidently reduces the flight time.

4.2.6. Analysis of the Mean Altitude

A further metric that indicates the differences between the three system architectures is the mean altitude at each time step over all simulation runs, i.e for both failed and successful flights, as illustrated in Figure 4.10 for all three architectures. Note that the peak at approx. 150 s and 1200 m is the point where the glider leaves the initial updraft at the origin of the coordinate system. First, the figure illustrates that the blind system architecture features a significant variation in altitude towards the target. The mean altitude is dominated by failed missions, causing outlandings after approximately 1500 s. This phenomenon could already be observed in the flight trajectory study, as illustrated on Figure 4.7a. In comparison, the two vision-based architectures maintain a higher average altitude. Furthermore, the altitude of the vision-based architectures remains almost constant between the first leg and the last leg. Also

note that the more conservative planning of the probabilistic planner is reflected in a slightly higher mean altitude, compared to the deterministic planner.

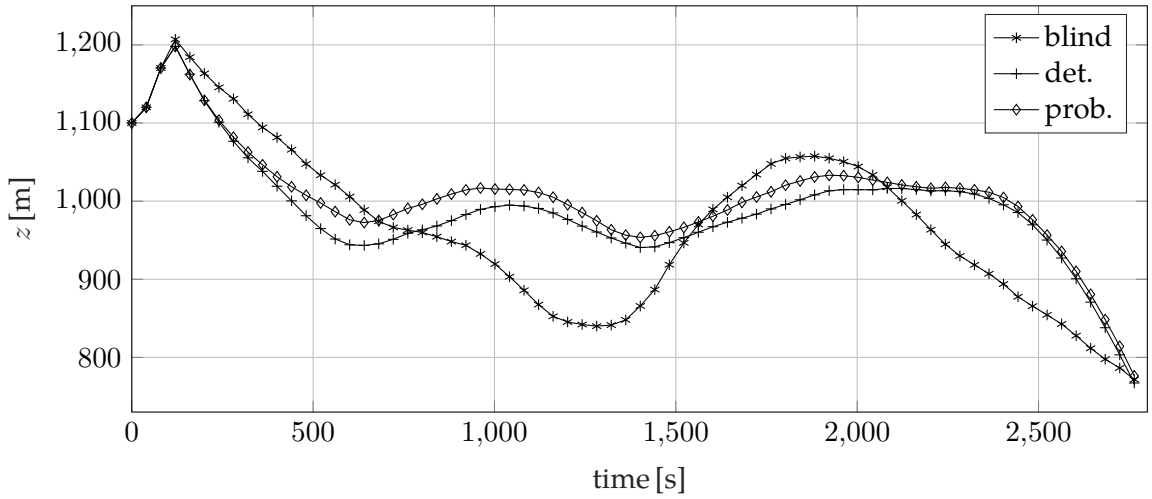


Figure 4.10: Average altitude

4.3. Cost Function Variations

In addition to the MC simulation and sensitivity study, further simulations were conducted to illustrate the impact of varying cost function settings of the proposed path planning method. Recall from Chapter 3 that the cost for transition to target actions is function of the two parameters k_0 and k_1. The first parameter serves to penalize non-feasible paths in general, and k_1 weights the cost of a path depending on the remaining distance from the outlanding site to the target. At the beginning of a flight, k_0 will typically have the same impact on all routes, since the destination is far away and not yet reachable. The parameter k_1 induces a more visible impact on the flight trajectories. For illustrating this effect, a flight with equal meteorological parameters was repeated three times, varying the setting of k_1, i.e. the distance-dependent term, while keeping k_0 at 5000 s. The parameter settings and the corresponding flight trajectories are illustrated in Figure 4.11. Note that according to Equation (3.42), the three values for k_1 can be transformed into conveyor belt velocities (Chapter 3) of 0.1 m/s, 1 m/s, and 10 m/s respectively, since $V_a^\star$ is 10 m/s for final glide archs that violate the minimum altitude constraint z_{min}.

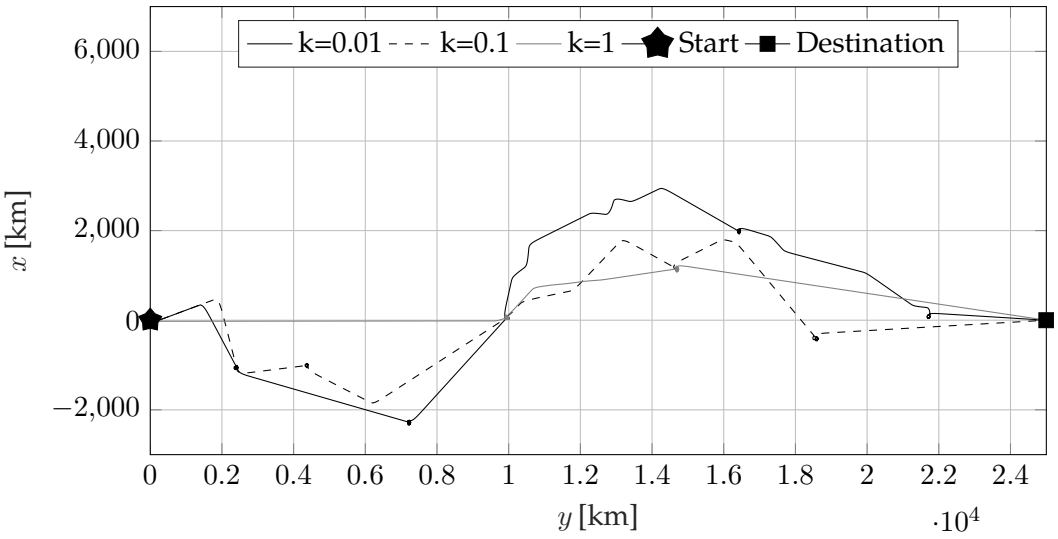


Figure 4.11: Trajectories depending on the distance-depending cost function parameter k_1

The black solid line shows the trajectory resulting from the most conservative parameter setting. With a high cost on the remaining distance, mission success is valued higher compared to flight time reduction. This can induce significant deviations from the direct course and thus increases the flight time which is 3092 s. For k_1=0.1, the course deviations are smaller, which results in a decreased flight time of 3036 s. The trajectory that results from a setting $k_1 = 1$

lies almost on the direct line between the start and the target and evidently yields the shortest flight time of 2784 s, i.e. approximately 5 min faster than with the most conservative setting. The impact of the parameter setting can also be observed by counting the number of updrafts that the vehicle approaches. In the first two cases, the vehicle approaches 5 updrafts (see the small dots that lie on the trajectories). Conversely, in the case of the most aggressive setting, only 2 updrafts are approached. Note that the flight was executed using system architecture 3, i.e. using vision and the probabilistic planner, but the same effects can be observed using the deterministic planner.

4.4. Summary

In this chapter, an overall evaluation of the proposed navigation and path planning methods for autonomous cross-country soaring is presented. For enabling both a detailed evaluation of the presented methods and highlighting the benefits compared to state-of-the-art approaches, three system architectures are established and tested, each of these relying on different updraft estimation and/or path planning methods. Furthermore, four Monte Carlo setups are defined covering two realistic updraft density levels and two image processing noise levels, additionally enabling sensitivity studies of the architectures w.r.t. the most crucial parameters. The Monte Carlo (MC) simulation results reveal the capacity of the proposed remote updraft estimation to significantly reduce the number of outlandings on typical cross-country soaring flights, even when only relying on a deterministic path planner. As expected, the simulation results demonstrate the capability of the proposed probabilistic path planner to further decrease the number of outlandings. Additionally, flight time statistics indicate that the probabilistic path planning approach yields acceptable increases of the average flight time to the destination, compared to the deterministic planner.

Chapter 5

Conclusion

In this chapter, the thesis' contributions are summarized, and directions for future research in this field are discussed.

Autonomous soaring for fixed-wing UAVs is a relatively new field of research, and a technique that offers significant potential to increase both the flight time and the range of unmanned aircraft. However, a key challenge in autonomous soaring lies in the identification of updrafts and in the generation of efficient flight paths.

Therefore, this thesis has proposed a framework for Vision-Based Autonomous Cross-Country Soaring (VBACCS). The originality of this work lies in the use of vision for remotely estimating the position and parameters of sub-cumulus thermal updrafts, thereby mimicking the behavior of human glider pilots. While the realization of a VBACCS system necessitates progress in various fields, the focus was put on the remote estimation and path planning part. Although the primary purpose of this thesis is autonomous soaring, some of the contributions find application beyond that field, e.g. in meteorological research.

5.1. Thesis Contributions

5.1.1. Vision-based Framework for Autonomous Cross-Country Soaring

The first contribution of this thesis is a new framework for reliable autonomous cross-country soaring flight of a small glider UAV that relies on vision for remotely estimating updrafts that are capped by cumulus clouds. The idea for the framework is inspired by the behavior of human glider pilots who scan the environment for cumulus clouds and use this information in their decision-making. Parts of this concept are realized by proposing algorithms for vision-based updraft estimation and map-based path planning complemented by an evaluation in a simulation environment.

5.1.2. Vision-based Remote Updraft Estimation

In this thesis, an estimator for remotely estimating updrafts is developed. Regarding the state of the art before this thesis, the estimator provides an original solution to the remote updraft estimation problem.

The UKF is identified as suitable candidate due to its capability of providing consistent state estimates for non-linear systems. The proposed estimator features a hybrid architecture, in which the estimation of updraft positions and updraft parameters are performed separately in order to cope with varying initialization dynamics.

Both vision measurements and the vehicle's attitude and position are taken as filter inputs in order to calculate the position of cumulus clouds in the first filter. The phenomenon of a uniform cloud base altitude was exploited by estimating cloud positions in one filter, rather than in individual filters in order to increase estimation convergence. Furthermore, a virtual cloud width sensor was designed. The sensor transforms vision measurements and the cloud's estimated position from the first filter into virtual measurements of the current cloud width. The technique of Unscented Transform (UT) is applied to reliably propagate measurement uncertainties through the system equations of the sensor. The proposed cloud width sensor can also be used for measurement campaigns in meteorological research.

In the second filter, the measurements of cloud widths are used for tracking their evolution over time. True measurements of cloud width evolutions were used for identifying a simple function that describes the cloud size dynamics. This function is relied on to predict both the vanish time and the updraft strength related to the cumulus cloud. MCSs were conducted

in order to study both the convergence and consistency characteristics of the proposed estimator using a realistic observer soaring flight trajectory. The results show the capacity of the proposed filter to provide both convergent and consistent estimates in spite of adverse observation conditions.

5.1.3. Robust Path Planning For Map-Based Autonomous Cross-Country Soaring

Map-based path planning for autonomous cross-country soaring deals with the calculation of a series of updrafts that the glider has to approach for regaining altitude before it can reach its destination. However, under real world conditions, the updraft maps are imperfect due to the presence of measurement errors. Therefore, a robust path planner for map-based autonomous cross-country soaring was developed. The originality of the path planner is that it explicitly handles the trade-off between reducing flight times and a reduction of the outlanding risk by considering updraft map uncertainties.
The framework of MDPs was identified as suitable candidate. The path planning problem was formalized in a computationally efficient way by using a simple flight dynamics model. Further, a final glide guidance law was desinged to minimize the flight time of the final leg, while respecting constraints on the minimum altitude. Using the flight dynamics model and the technique of Unscented Transform (UT), the cross-country soaring problem can be modeled as a finite horizon MDP. A cost function based on a glider-competition rule was suggested for the calculation of transition costs and backwards induction was applied to determine the seeked optimal policy. A series of simulations illustrate the benefits of the proposed path planner in comparison to a state-of-the-art deterministic planner. Additionally, a study of the execution time of a prototype implementation is provided, indicating its implementability on-board a small scale UAV glider, equipped with low-cost computer devices.

5.1.4. Evaluation of the Overall Vision-Based Autonomous Cross-Country Soaring (VBACCS) System

To assess the joint performance of both the vision-based remote updraft estimation and the robust path planner, a series of Monte Carlo Simulation (MCS) was conducted using a realistic 6 DOF flight simulator including low-level flight controllers and a thermal centering controller. Three different system architectures for autonomous cross-country soaring were defined and

compared. One of these reproduces the state of the art in autonomous cross-country soaring, prior to this thesis. Thereby, performance of the proposed navigation and planning algorithms is not only individually assessed, but also compared. Representative meteorological parameter settings and varying image processing noise levels were used in the simulations in order to also study the sensibility of the proposed system with respect to crucial input parameters. The simulation results confirmed the significant potential of the proposed VBACCS system to contribute to a reduction of outlandings, while yielding only minor increases in flight time.

5.2. Limitations and Suggested Future Work

Working over a long time on a complex subject generates not only solutions, but also reveals questions and even more challenges. This section discusses the limitations of the presented work and suggests directions for future work.

- **Horizontal Wind:** In the presence of horizontal wind, the position of clouds and updrafts might not be stationary but change with time. To consider wind in the estimation process, the glider could be equipped with a wind estimator and the updraft estimator could rely on the wind estimates in the state prediction function. Another possibility would be to estimate both wind and cloud positions in one filter. Therefore, airdata (e.g. airspeed and angle of attack) and vision measurements would have to be fused. Moreover, models should be integrated that incorporate possible updraft tilting and bending due to horizontal wind. In the same vein, the effects of horizontal wind will have to be integrated in the state propagation model of the suggested path planner.

- **Attitude Uncertainties:** In this thesis, both attitude and position estimates of the vehicle were assumed to be of high quality. Degraded attitude estimates however propagate into the updraft estimation and can sensitively impact the estimation performance. An extension of the proposed estimator would therefore be to incorporate the uncertainty of attitude and vehicle position estimates in the corresponding system equations.

- **Position Uncertainties:** In this work, updraft position uncertainties were not considered in the path planning. However, especially in the presence of horizontal wind, increased updraft position uncertainties can be expected. This can impact the seek-time under an approached cloud. Further research should therefore concentrate on how these position uncertainties can efficiently be integrated into the proposed path planner, while keeping

the problem tractable.

- **Image-In-The-Loop-Simulation:** Another important direction of research will include the evaluation of the estimator with images in the loop. In a first step, synthetic images that reproduce realistic lightning and occlusion effects could be used. This requires a suited image generator. A possible direction would be to rely on images from a computer game flight simulator with a high degree of realism. Both the estimator and the path planner could thereby be tested in "closed-loop".
  Going one step further, flight data recordings and vision measurements should be collected for evaluating the estimator with true inputs. These inputs can for an example be collected using a manned motor glider plane.

- **Joint Airspeed and Updraft Selection:** In the proposed path planner, updraft and airspeed selection were separated. This procedure allows to keep the planning problem tractable, but also has its shortcomings, as was shown in Chapter 4. Therefore, future research should concentrate on how to efficiently combine the optimization of both the updraft and the airspeed selection.

- **A Priori Knowledge And Fusion Of Soaring Techniques:** Complementary to information on remotely located updrafts, it is also beneficial to incorporate geographical a priori knowledge in the path planning task. It can for an example be unfavorable to fly over open waters such as lakes or seas, since it is more likely to encounter temperature gradients and thus updrafts over land. In the same vein, it could be promising to jointly benefit from different soaring techniques (e.g. ridge soaring, dynamic soaring etc.) during one autonomous soaring flight. Therefore, the path planner could be extended to rely on geographical information together with the perceived environment in order to decide when to exploit which energy source and to set the corresponding mode of operation.

- **Adaptive Cost Function:** The proposed path planner relies on a cost function that incorporates the notion of an outlanding. Throughout this thesis, the cost function parameters for outlandings were kept constant during flight. Meteorological conditions in terms of both the updraft density and strength can vary during the course of the day. While updrafts are comparably weak in the morning and the early evening, stronger updrafts are encountered during the afternoon. For long distance flights, lasting several hours, the cost function of the proposed path planner could therefore be adapted based upon the current meteorological conditions. To give an example, if it is likely that updrafts will

occur on the future flight path, one can include this information by changing the cost function to be less conservative.

- **Target Hardware Implementation and Flight Testing:** Another important aspect is the implementation of the proposed algorithms on the designated hardware components. While prototypes of the systems were programmed for simulation purposes, code efficiency plays an important role for embedded systems, especially when relying on the computationally demanding framework of MDPs. Therefore, these prototypes should be optimized. Once the navigation and path planning components have been implemented, the entire system can be evaluated in real flight tests. Flight tests require both suitable meteorological conditions, as well as flight clearances, since the UAV glider will most likely fly outside the line of sight of the operator. Therefore, it could be interesting to conduct these flight test campaigns over uninhabited and desert-like regions such as the Mojave desert in California or the Namibia desert, thereby facilitating flight clearances while also benefiting from advantageous meteorological conditions that help to evaluate the algorithms.

Appendix A

UAV Glider Model

This section introduces the flight dynamics model of a small scale glider plane that is used in Chapters 2 to 4 to generate representative flight trajectories.

A.1. Aircraft Model

This work was supported by MULTIPLEX® Modellsport GmbH & Co. KG, Germany, by providing two fully equipped Cularis electric gliders for video data collection and future flight tests. The main characteristics of the glider were identified in order to provide a realistic glider dynamics model for computer simulations.
The geometric, inertial, and mass properties of the Cularis glider are noted in Table A.1.

Parameter	Symbol	Value	Unit
Mass	m	2	[kg]
Wingspan	b	2.61	[m]
Wing surface	S	0.42	[m^2]
Mean aerodynamic chord	$\bar{c}$	0.17	[m]
Inertia tensor	I	$\begin{bmatrix} 0.1686 & 0 & -0.0055 \\ 0 & 0.1300 & 0 \\ -0.0055 & 0 & 0.2964 \end{bmatrix}$	[kgm^2]

Table A.1: Main properties of the Cularis glider

A.1.1. Aerodynamic Forces and Moments

The aerodynamic forces $\mathbf{F}_a$ and moments $\mathbf{M}_a$ (in the body frame) that act on the vehicle are defined as

$$\begin{aligned}\mathbf{F}_a &= \mathbf{R}_{b\leftarrow s}\bar{q}S\begin{bmatrix}C_D\\C_Y\\C_L\end{bmatrix},\\ \mathbf{M}_a &= \mathbf{R}_{b\leftarrow s}\begin{bmatrix}\bar{q}Sb & 0 & 0\\0 & \bar{q}S\bar{c} & 0\\0 & 0 & \bar{q}Sb\end{bmatrix}\begin{bmatrix}C_l\\C_m\\C_n\end{bmatrix},\end{aligned} \tag{A.1}$$

where C_D, C_Y, C_L and C_l, C_m, C_n are the dimensionless aerodynamic coefficients for forces and moments respectively. The dynamic pressure is defined as $\bar{q} = 0.5\rho V_a^2$ where ρ specifies the air density. The matrix $\mathbf{R}_{b\leftarrow s}$ conducts transformations between the stability frame and the body frame of the vehicle. Note that the stability frame s is similar to the body-fixed reference frame b. The stability frame is obtained when rotating the body frame about the Y_b axis by the angle of attack α.

Specifying the rotation rates $\mathbf{\Omega} = [p\ q\ r]^T$ and translational velocities $\mathbf{V} = [u\ v\ w]^T$ in the body frame, both angular and translational accelerations are defined as

$$\begin{aligned}\dot{\mathbf{\Omega}} &= I^{-1}\left(\mathbf{M} - \mathbf{\Omega}\times I\mathbf{\Omega}\right) = I^{-1}\left(\mathbf{M}_a - \mathbf{\Omega}\times I\mathbf{\Omega}\right),\\ \dot{\mathbf{V}} &= \frac{\mathbf{F}}{m} - \mathbf{\Omega}\mathbf{V},\end{aligned} \tag{A.2}$$

where

$$\mathbf{F} = \mathbf{F}_a + \mathbf{F}_g = \mathbf{F}_a + \mathbf{R}_{b\leftarrow in}\begin{bmatrix}0\\0\\mg\end{bmatrix} \tag{A.3}$$

and I represents the inertia tensor according to

$$I = \begin{bmatrix}I_{xx} & 0 & I_{xz}\\0 & I_{yy} & 0\\I_{xz} & 0 & I_{zz}\end{bmatrix}. \tag{A.4}$$

Adopting the notation from [84], the coefficients for aerodynamic forces and moments as introduced in Section A.1.1 are obtained as the sum of the datum-coefficients, the actuator in-

crements, and the body rate damping increments

$$
\begin{aligned}
C_D &= C_{D_0} + (C_{L_0} + C_{L_\alpha}\alpha)^2 \frac{1}{\pi e \Lambda} + C_{D_{\delta_e}} + C_{D_q}\frac{\bar{c}}{2V_a}q, \\
C_Y &= C_{Y_\beta}\beta + C_{Y_p}\frac{b}{2V_a}p + C_{Y_r}\frac{b}{2V_a}r + C_{Y_{\delta_a}}\delta_a + C_{Y_{\delta_r}}\delta_r, \\
C_L &= C_{L_0} + C_{L_\alpha}\alpha + C_{L_q}\frac{\bar{c}}{2V_a}q + C_{L_{\delta_e}}\delta_e, \\
C_l &= C_{l_0} + C_{l_\beta}\beta + C_{l_p}\frac{b}{2V_a}p + C_{l_r}\frac{b}{2V_a}r + C_{l_{\delta_a}}\delta_a + C_{l_{\delta_r}}\delta_r, \\
C_m &= C_{m_0} + C_{m_\alpha}\alpha + C_{m_q}\frac{\bar{c}}{2V_a}q + C_{m_{\delta_e}}\delta_e, \\
C_n &= C_{n_0} + C_{n_\beta}\beta + C_{n_p}\frac{b}{2V_a}p + C_{n_r}\frac{b}{2V_a}r + C_{n_{\delta_a}}\delta_a + C_{n_{\delta_r}}\delta_r,
\end{aligned}
\tag{A.5}
$$

where e is the Oswald factor and Λ stands for the wing aspect ratio. The actuator deflections for aileron, elevator, and rudder are denoted δ_a, δ_e, and δ_r respectively.

The AVL software, as introduced by Mark Drela at the MIT, is used for determining the aerodynamic coefficients of the Cularis glider based on the characteristic geometric properties of the vehicle. Therefore, the geometric properties of the Cularis were measured. The 3D-AVL geometry of the Cularis glider is illustrated in Figure A.1. The identified coefficients are listed in Tables A.2 to A.5.

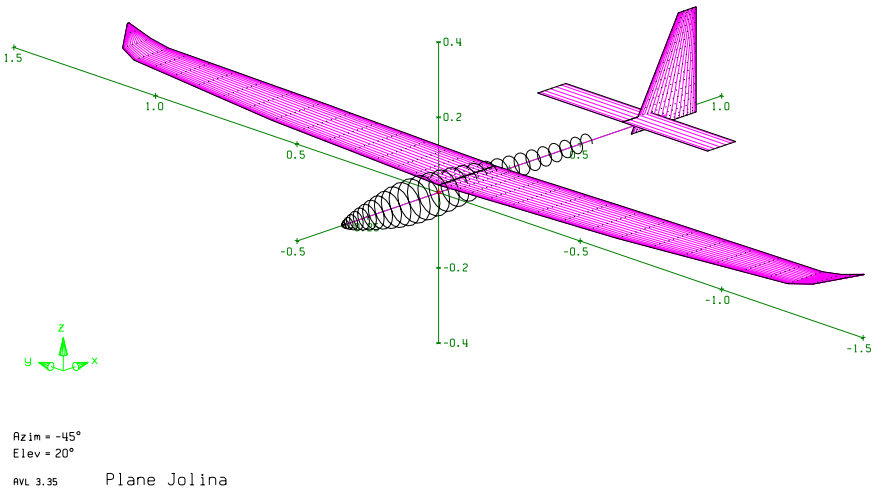


Figure A.1: AVL airframe geometry plot of the Cularis glider

Force/Moment	w.r.t. α		w.r.t. β	
L	$C_{L_\alpha} =$	6.271	$C_{L_\beta} =$	0.000
Y	$C_{Y_\alpha} =$	0.000	$C_{Y_\beta} =$	-0.242
l	$C_{l_\alpha} =$	0.000	$C_{l_\beta} =$	-0.072
m	$C_{m_\alpha} =$	-1.402	$C_{m_\beta} =$	0.000
n	$C_{n_\alpha} =$	0.000	$C_{n_\beta} =$	0.052

Table A.2: Datum coefficients

Force/Moment	w.r.t. p		w.r.t. q		w.r.t. r	
L	$C_{L_p} =$	0.000	$C_{L_q} =$	10.600	$C_{L_r} =$	0.000
Y	$C_{Y_p} =$	0.016	$C_{Y_q} =$	0.000	$C_{Y_r} =$	0.1411
l	$C_{l_p} =$	-0.634	$C_{l_q} =$	0.000	$C_{l_r} =$	0.1577
m	$C_{m_p} =$	0.000	$C_{m_q} =$	-19.446	$C_{m_r} =$	0.000
n	$C_{n_p} =$	-0.058	$C_{n_q} =$	0.000	$C_{n_r} =$	-0.042

Table A.3: Rate coefficients

Force/Moment	w.r.t. δ_a		w.r.t. δ_e		w.r.t. δ_r	
L	$C_{L_{\delta_a}} =$	0.000	$C_{L_{\delta_e}} =$	0.773	$C_{L_{\delta_r}} =$	0.000
Y	$C_{Y_{\delta_a}} =$	-0.012	$C_{Y_{\delta_e}} =$	0.000	$C_{Y_{\delta_r}} =$	0.187
l	$C_{l_{\delta_a}} =$	-0.339	$C_{l_{\delta_e}} =$	0.000	$C_{l_{\delta_r}} =$	0.006
m	$C_{m_{\delta_a}} =$	0.000	$C_{m_{\delta_e}} =$	-2.475	$C_{m_{\delta_r}} =$	0.000
n	$C_{n_{\delta_a}} =$	0.017	$C_{n_{\delta_e}} =$	0.000	$C_{n_{\delta_r}} =$	-0.054

Table A.4: Actuator coefficients

Force/Moment	Coefficient	
L	C_{L_0}=	0.63
D	C_{D_0}=	0.0254
l	C_{l_0}=	0
m	C_{m_0}=	-0.085
n	C_{n_0}=	0

Table A.5: Zero coefficients

A.1.2. The Polar Curve

As discussed in Chapter 3, the polar curve is a fundamental aerodynamic characteristic for glider planes. It relates the airspeed V_a to the sink rate v_{si} of the vehicle and is well approximated by a second order function such that

$$v_{si} = aV_a^2 + bV_a + c. \tag{A.6}$$

Note that the sink rate is defined positive downwards. The identified polar curve for the Cularis glider is illustrated in Figure A.2 where the point of contact defined by the tangent from the origin and the polar curve specifies the vehicle's best glide ratio. In the simulation, the airspeed of the best glide is set to 10 m/s.

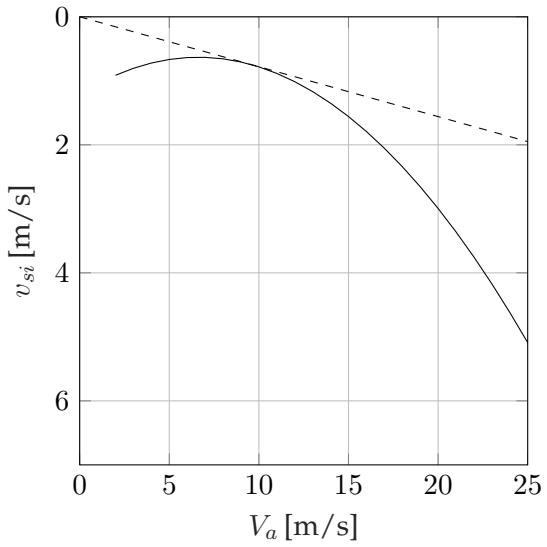


Figure A.2: Cularis polar curve with $a = 0.0132$, $b = -0.1748$, $c = 1.21$

Appendix B

Flight Control Laws

A set of low level control laws is designed in order to stabilize the Cularis glider from Appendix A. While there exists a variety of advanced controller design methods, here, the purpose of the controllers is simply to generate a realistic glider motion in the simulation and to enable the vehicle to track high-order commands that are generated from both the guidance and the path planning. These commands include the airspeed for the longitudinal motion and the bank angle for the lateral motion.
In the following section, the structure of these controllers is shortly introduced, and the calculated gains are noted.

B.1. Bank Angle and Airspeed Control

The control loops for both bank angle and airspeed were designed using the technique of Eigenstructure Assignment, as explained by the authors of [85]. The corresponding control structures are noted in Equations (B.1) and (B.2) respectively.
Note that the controller gains are scaled with the dynamic pressure $\bar{q}$. This procedure enables to obtain quasi-equal closed-loop performance across the flight envelope. The reference dynamic pressure $\bar{q}_{ref} = 48.33\,\mathrm{kg/(ms^2)}$ indicates the nominal value at which the linear controllers were designed.
For the bank angle controller, the control error $\Phi_{cmd} - \Phi$ and its integral, as well as the roll rate

p are fed back in order to stabilize the aircraft's roll mode.

$$\delta_a = \underbrace{\begin{bmatrix} -1.0751 & -0.9405 & 0.0747 \end{bmatrix}}_{K_\Phi} \begin{bmatrix} \Phi_{cmd} - \Phi \\ \int \Phi_{cmd} - \Phi \, d_t \\ p \end{bmatrix} \frac{\bar{q}_{ref}}{\bar{q}}. \tag{B.1}$$

Note that all angles are given in rad.

The airspeed controller Equation (B.2) generates appropriate short period mode damping around values of 0.7. Both the airspeed error $V_{a,cmd} - V_a$ and the integrated airspeed error as well as the angle of attack α, the pitch angle Θ, and the pitch rate q are fed back.

$$\delta_e = \underbrace{\begin{bmatrix} 0.013 & 0.006 & 0.002 & -0.219 & 0.005 \end{bmatrix}}_{K_{V_a}} \begin{bmatrix} V_{a,cmd} - V_a \\ \int V_{a,cmd} - V_a \, dt \\ \alpha_{trim} - \alpha \\ \Theta_{trim} - \Theta \\ q \end{bmatrix} \frac{\bar{q}_{ref}}{\bar{q}} \tag{B.2}$$

with

$$\alpha_{trim} = 0.0335, \Theta_{trim} = -0.0175. \tag{B.3}$$

B.2. Yaw Damper

For increasing the damping of the dutch roll mode, and also reducing parasitic side slip angles β arising when banking the vehicle, a yaw damper Equation (B.4) is designed, actuating the rudder. A washout filter with a time constant of 2 s is used for high-pass filtering the yaw rate r which is fed back to the rudder. It avoids counteracting the desired rate during a turn. A feed forward control $\Phi_{cmd} - \Phi$ is added to achieve faster responses.

$$\delta_r = \underbrace{\begin{bmatrix} 0.75 & -1 \end{bmatrix}}_{K_{yd}} \begin{bmatrix} r_{wo} \\ \Phi_{cmd} - \Phi \end{bmatrix} \frac{\bar{q}_{ref}}{\bar{q}}, \tag{B.4}$$

where r_{wo} specifies the filtered yaw rate r.

B.3. Thermal Centering Controller

Note that the functionality of the thermal centering controller has been explained in Chapter 4. Following Equations (4.2) and (4.3), bank angle commands are generated according to

$$\phi_{cmd} = \arctan\left(\frac{V_a}{g\dot{\Psi}_{cmd}}\right) \tag{B.5}$$

and the commanded turn rate is a function stated as

$$\dot{\Psi}_{cmd} = \underbrace{\frac{V_a}{40}}_{r_{ss}} - \underbrace{1.2}_{K_{tc}}\ddot{E}. \tag{B.6}$$

The steady state radius r_{ss} and the gain K_{tc} specify the control law gains of the thermal centering law.

Appendix C

Bounding Box Calculation

Let $\mathbf{x} = [x_1\, x_2]^T$ be a random vector such that

$$\mathbf{x} \sim \mathcal{N}\left(\bar{\mathbf{x}}, \mathbf{P}\right), \tag{C.1}$$

with mean $\bar{\mathbf{x}}$ and covariance $\mathbf{P}$. Further, let $\boldsymbol{\lambda} = [\lambda_1\ \lambda_2]^T$ specify the two eigenvalues that are related to the covariance matrix $\mathbf{P}$. Given a scaling factor ξ, as well as the eigenvalues $\boldsymbol{\lambda}$, we can specify the semi-minor a_{el} and semi-major b_{el} axis of the related ellipse such that

$$\begin{aligned} a_{el} &= \sqrt{\xi \max\left(\boldsymbol{\lambda}\right)} \\ b_{el} &= \sqrt{\xi \min\left(\boldsymbol{\lambda}\right)}. \end{aligned} \tag{C.2}$$

The scaling factor is related to the selected confidence level 1-α. More specifically, the scale factor ξ is defined by

$$\xi = F^{-1}_{\chi^2_2}\left(1-\alpha\right). \tag{C.3}$$

The rotation angle Θ of the ellipse w.r.t. the coordinate frame is

$$\Theta = \arctan\left(\frac{\max\left(\boldsymbol{\lambda}\right)}{\min\left(\boldsymbol{\lambda}\right)}\right). \tag{C.4}$$

Then, the bounds of the ellipse's surrounding box are

$$\begin{aligned} \mathbf{x}_{\bar{1},\underline{1}} &= \bar{\mathbf{x}}_1 \pm \sqrt{a_{el}^2 \cos^2\Theta + b_{el}^2 \sin^2\Theta} \\ \mathbf{x}_{\bar{2},\underline{2}} &= \bar{\mathbf{x}}_2 \pm \sqrt{a_{el}^2 \sin^2\Theta + b_{el}^2 \cos^2\Theta}. \end{aligned} \tag{C.5}$$

Appendix D

Bivariate Conditional Expectation

Consider the two dimensional random variable from Appendix C. Further, let $f_{\mathbf{x}}(\mathbf{x})$ be the probability density function of the bivariate normal distribution according to

$$f_{\mathbf{x}}(\mathbf{x}) = \frac{1}{2\pi|\mathbf{P}|^{\frac{1}{2}}} e^{-\frac{1}{2}\left((\mathbf{x}-\bar{\mathbf{x}})^T \mathbf{P}^{-1}(\mathbf{x}-\bar{\mathbf{x}})\right)}. \tag{D.1}$$

Let $\mathcal{B}$ specify a rectangular region, aligned with the coordinate frame defined by x_1 and x_2.

$$\mathcal{B} = \{\mathbf{x} | x_{1,min} \leq x_1 \leq x_{1,max}, x_{2,min} \leq x_2 \leq x_{2,max}\}. \tag{D.2}$$

Then, the conditional expectation $\mathbb{E}(\mathbf{x}|\mathbf{x} \in \mathcal{B})$ is defined as

$$\mathbb{E}(\mathbf{x}|\mathbf{x} \in \mathcal{B}) = \frac{1}{P(\mathbf{x} \in \mathcal{B})} \int_{\mathcal{B}} \mathbf{x} f_{\mathbf{x}}(\mathbf{x})\, d\mathbf{x}. \tag{D.3}$$

Assume the probability $P(\mathbf{x} \in \mathcal{B})$ is known, for example calculated using the methods provided in [79]. Then, the double integral in Equation (D.3) has to be solved in order to determine the conditional expectation.

Assume the covariance matrix $\mathbf{P}$ can be diagonalized as

$$\mathbf{P} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T, \tag{D.4}$$

where $\mathbf{V}$ is the eigenvector matrix, and $\mathbf{\Lambda}$ is the diagonal eigenvalue matrix, both specified as

$$\begin{aligned} \mathbf{V} &= \begin{bmatrix} v_{11} & v_{12} \\ v_{21} & v_{22} \end{bmatrix}, \\ \mathbf{\Lambda} &= \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}. \end{aligned} \tag{D.5}$$

Define a transformed vector $\tilde{\mathbf{x}}$ according to

$$\tilde{\mathbf{x}} = \mathbf{V}^T \left(\mathbf{x} - \bar{\mathbf{x}} \right). \tag{D.6}$$

Using the transformation Equation (D.6), a transformed rectangle $\tilde{\mathcal{B}} = \mathbf{V}^T(\mathcal{B} - \bar{\mathbf{x}})$ can be generated, as illustrated in Figure D.1. The transformed region can be separated in at most three subregions $\tilde{\mathcal{B}}_1, \tilde{\mathcal{B}}_2$ and $\tilde{\mathcal{B}}_3$, as further illustrated in Figure D.1. For each subregion, we can establish lower and upper bounds on $\tilde{x}_2$, noted $\tilde{x}_{2_{l,i}}$ and $\tilde{x}_{2_{u,i}}$ respectively. These bounds are illustrated in Figure D.1 for $\tilde{\mathcal{B}}_2$.

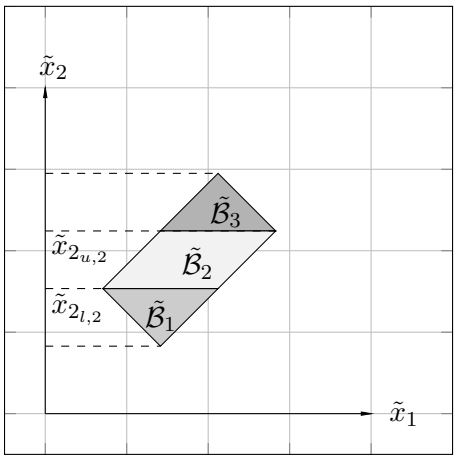


Figure D.1: Region decomposition in the transformed frame

For each subregion, we can also specify two linear functions $g_{l,i}$ and $g_{u,i}$ where the functions describe the outer bounds of the subregion

$$g_{l,i}\left(\tilde{x}_2\right) = \alpha_{l,i}\tilde{x}_2 + \beta_{l,i}, \tag{D.7}$$

$$g_{u,i}\left(\tilde{x}_2\right) = \alpha_{u,i}\tilde{x}_2 + \beta_{u,i}. \tag{D.8}$$

If we know the conditional expectation of the transformed variable, we can propagate it back in order to obtain the conditional expectation in the original frame

$$\mathbb{E}\left[\mathbf{x}|\mathbf{x}\in\mathcal{B}\right]=\mathbf{V}\mathbb{E}\left[\tilde{\mathbf{x}}|\tilde{\mathbf{x}}\in\tilde{\mathcal{B}}\right]+\bar{\mathbf{x}}. \tag{D.9}$$

The conditional expectation of the transformed variable $\tilde{\mathbf{x}}$ is defined by

$$\mathbb{E}\left[\tilde{\mathbf{x}}|\tilde{\mathbf{x}}\in\tilde{\mathcal{B}}\right]=\frac{1}{P\left(\tilde{\mathbf{x}}\in\tilde{\mathcal{B}}\right)}\sum_{i=1}^{3}\left[\begin{array}{c}\underbrace{\frac{1}{2\pi|\Lambda|^{\frac{1}{2}}}\int_{\tilde{x}_{2_{l,i}}}^{\tilde{x}_{2_{u,i}}}e^{-\frac{1}{2}\frac{\tilde{x}_2^2}{\lambda_2}}\int_{\alpha_{l,i}\tilde{x}_2+\beta_{l,i}}^{\alpha_{u,i}\tilde{x}_2+\beta_{u,i}}\tilde{x}_1e^{-\frac{1}{2}\frac{\tilde{x}_1^2}{\lambda_1}}d\tilde{x}_1d\tilde{x}_2}_{c_{1,i}}\\ \underbrace{\frac{1}{2\pi|\Lambda|^{\frac{1}{2}}}\int_{\tilde{x}_{2_{l,i}}}^{\tilde{x}_{2_{u,i}}}\tilde{x}_2e^{-\frac{1}{2}\frac{\tilde{x}_2^2}{\lambda_2}}\int_{\alpha_{l,i}\tilde{x}_2+\beta_{l,i}}^{\alpha_{u,i}\tilde{x}_2+\beta_{u,i}}e^{-\frac{1}{2}\frac{\tilde{x}_1^2}{\lambda_1}}d\tilde{x}_1d\tilde{x}_2}_{c_{2,i}}\end{array}\right]. \tag{D.10}$$

It is now shown how the two integrals c_1 and c_2 can be solved analytically.
For c_1, integration over $\tilde{x}_1$ and rearranging yields

$$\begin{aligned}c_1=&\frac{-\lambda_1}{2\pi|\Lambda|^{\frac{1}{2}}}\int_{\tilde{x}_{2,l}}^{\tilde{x}_{2,u}}e^{-\frac{\beta_u^2}{2\left(\lambda_1+\lambda_2\alpha_u^2\right)}}e^{-\frac{\lambda_1+\lambda_2\alpha_u^2}{2\lambda_1\lambda_2}\left(\tilde{x}_2+\frac{\alpha_u\beta_u\lambda_2}{\lambda_1+\lambda_2\alpha_u^2}\right)^2}d\tilde{x}_2\\&-\int_{\tilde{x}_{2,l}}^{\tilde{x}_{2,u}}e^{-\frac{\beta_l^2}{2\left(\lambda_1+\lambda_2\alpha_l^2\right)}}e^{-\frac{\lambda_1+\lambda_2\alpha_l^2}{2\lambda_1\lambda_2}\left(\tilde{x}_2+\frac{\alpha_l\beta_l\lambda_2}{\lambda_1+\lambda_2\alpha_l^2}\right)^2}d\tilde{x}_2.\end{aligned} \tag{D.11}$$

By applying substitution, we can rewrite the term such that

$$c_1=-\frac{\lambda_1}{2\pi|\Lambda|^{\frac{1}{2}}}\left(e^{-\frac{\beta_u^2}{2\left(\lambda_1+\lambda_2\alpha_u^2\right)}}\int_{\tilde{t}_l}^{\tilde{t}_u}e^{-t^2}d\tilde{t}e^{-\frac{\beta_l^2}{2\left(\lambda_1+\lambda_2\alpha_l^2\right)}}\int_{\tilde{s}_l}^{\tilde{s}_u}e^{-s^2}d\tilde{s}\right). \tag{D.12}$$

Integration over the substituted variables $\tilde{t}$ and $\tilde{s}$, and rearranging yields

$$c_1=-\frac{\lambda_1}{2\sqrt{2\pi}}\left(k_{1,u}\left(\operatorname{erf}\left(\tilde{t}_u\right)-\operatorname{erf}\left(\tilde{t}_l\right)\right)-k_{1,l}\left(\operatorname{erf}\left(\tilde{s}_u\right)-\operatorname{erf}\left(\tilde{s}_l\right)\right)\right), \tag{D.13}$$

where the corresponding bounds on the new variables are

$$\begin{aligned}\tilde{t}_{u/l}=&\sqrt{\frac{\lambda_1+\lambda_2\alpha_u^2}{2\lambda_1\lambda_2}}\left(\tilde{x}_{2_{u/l}}+\frac{\lambda_2\alpha_u\beta_u}{\lambda_1+\lambda_2\alpha_u^2}\right),\\ \tilde{s}_{u/l}=&\sqrt{\frac{\lambda_1+\lambda_2\alpha_l^2}{2\lambda_1\lambda_2}}\left(\tilde{x}_{2_{u/l}}+\frac{\lambda_2\alpha_l\beta_l}{\lambda_1+\lambda_2\alpha_l^2}\right)\end{aligned} \tag{D.14}$$

and the parameters $k_{1_{u/l}}$ are defined as

$$\begin{aligned} k_{1,u} =& \frac{e^{-\frac{\beta_u^2}{2\left(\lambda_1+\lambda_2\alpha_u^2\right)}}}{\sqrt{\lambda_1+\lambda_2\alpha_u^2}}, \\ k_{1,l} =& \frac{e^{-\frac{\beta_l^2}{2\left(\lambda_1+\lambda_2\alpha_l^2\right)}}}{\sqrt{\lambda_1+\lambda_2\alpha_l^2}}. \end{aligned} \tag{D.15}$$

Note that "erf" specifies the error function which is defined according to

$$\text{erf}\,(x)\,\frac{1}{\sqrt{\pi}} = \int_{-x}^{x} e^{-t^2}\,dt. \tag{D.16}$$

The parameter c_2 is obtained in a similar manner. Integration over the inner variable $\tilde{x}_1$ results in

$$c_2 = \frac{1}{2\sqrt{\pi}\sqrt{2}\sqrt{\lambda_1}} \int_{\tilde{x}_{2,l}}^{\tilde{x}_{2,u}} \tilde{x}_2 e^{-\frac{\tilde{x}_2^2}{2\lambda_2}} \left(\text{erf}\left(\frac{\alpha_u\tilde{x}_2+\beta_u}{\sqrt{2\lambda_1}}\right) - \text{erf}\left(\frac{\alpha_l\tilde{x}_2+\beta_l}{\sqrt{2\lambda_1}}\right)\right) d\tilde{x}_2. \tag{D.17}$$

Using the formula for integration by parts, we can write

$$\begin{aligned} c_2 = \frac{1}{\sqrt{2}\sqrt{\pi}\sqrt{\lambda_2}} \Bigg(& \left[e^{\frac{\tilde{x}_2}{2\lambda_2}} \left(\text{erf}\left(\frac{\alpha_u\tilde{x}_2+\beta_u}{\sqrt{2\lambda_1}}\right) - \text{erf}\left(\frac{\alpha_l\tilde{x}_2+\beta_l}{\sqrt{2\lambda_1}}\right)\right)\right]_{x_{2,l}}^{x_{2,u}} \\ & - \int_{\tilde{x}_{2,l}}^{\tilde{x}_{2,u}} e^{-\frac{\tilde{x}_2^2}{2\lambda_2}} \frac{\sqrt{2}}{\sqrt{\pi}} \left(e^{-\frac{\left(\alpha_u\tilde{x}_2+\beta_u\right)^2}{2\lambda_1}} \frac{\alpha_u}{\sqrt{2\lambda_1}} - e^{-\frac{\left(\alpha_l\tilde{x}_2+\beta_l\right)^2}{2\lambda_1}} \frac{\alpha_l}{\sqrt{2\lambda_1}} \right) d\tilde{x}_2 \Bigg). \end{aligned} \tag{D.18}$$

Rearranging the terms, we obtain

$$\begin{aligned} c_2 = & \frac{1}{\sqrt{2}\sqrt{\pi}\sqrt{\lambda_2}} \left(\left[e^{\frac{\tilde{x}_2}{2\lambda_2}} \left(\text{erf}\left(\frac{\alpha_u\tilde{x}_2+\beta_u}{\sqrt{2\lambda_1}}\right) - \text{erf}\left(\frac{\alpha_l\tilde{x}_2+\beta_l}{\sqrt{2\lambda_1}}\right)\right)\right]_{x_{2,l}}^{x_{2,u}} \right) \\ & + \frac{\sqrt{\lambda_2}}{2\pi\sqrt{\lambda_1}} \Bigg(e^{-\frac{\beta_u^2}{2\left(\lambda_1+\lambda_2\alpha_u^2\right)}} \alpha_u \int_{\tilde{x}_{2,l}}^{\tilde{x}_{2,u}} e^{-\frac{\lambda_1+\lambda_2\alpha_u^2}{2\lambda_1\lambda_2}\left(\tilde{x}_2\frac{\lambda_2\alpha_u\beta_u}{\lambda_1+\lambda_2\alpha_u^2}\right)^2} d\tilde{x}_2 \\ & - e^{-\frac{\beta_l^2}{2\left(\lambda_1+\lambda_2\alpha_l^2\right)}} \alpha_l \int_{\tilde{x}_{2,l}}^{\tilde{x}_{2,u}} e^{-\frac{\lambda_1+\lambda_2\alpha_l^2}{2\lambda_1\lambda_2}\left(\tilde{x}_2+\frac{\lambda_2\alpha_l\beta_l}{\lambda_1+\lambda_2\alpha_l^2}\right)^2} d\tilde{x}_2 \Bigg). \end{aligned} \tag{D.19}$$

Substitution, and integration over the substituted variables results in

$$
\begin{aligned}
c_2 = & \frac{1}{\sqrt{2}\sqrt{\pi}\sqrt{\lambda_2}} \left(\left[e^{\frac{\tilde{x}_2}{2\lambda_2}} \left(\operatorname{erf}\left(\frac{\alpha_u \tilde{x}_2 + \beta_u}{\sqrt{2\lambda_1}} \right) - \operatorname{erf}\left(\frac{\alpha_l \tilde{x}_2 + \beta_l}{\sqrt{2\lambda_1}} \right) \right) \right]_{x_{2,l}}^{x_{2,u}} \right) \\
& + \frac{\lambda_2}{2\sqrt{2}} \left(\alpha_u \frac{e^{-\frac{\beta_u^2}{2\left(\lambda_1 + \lambda_2 \alpha_u^2\right)}}}{\sqrt{\lambda_1 + \lambda_2 \alpha_u^2}} \left(\operatorname{erf}\left(\tilde{t}_u \right) - \operatorname{erf}\left(\tilde{t}_l \right) \right) - \alpha_l^2 \frac{e^{-\frac{\beta_l^2}{2\left(\lambda_1 + \lambda_2 \alpha_l^2\right)}}}{\sqrt{\lambda_1 + \lambda_2 \alpha_l^2}} \left(\operatorname{erf}\left(\tilde{s}_u \right) - \operatorname{erf}\left(\tilde{s}_l \right) \right) \right).
\end{aligned}
\tag{D.20}
$$

Defining

$$
\begin{aligned}
k_{2,u} &= \frac{e^{-\frac{\beta_u^2}{2\left(\lambda_1 + \lambda_2 \alpha_u^2\right)}}}{\sqrt{\lambda_1 + \lambda_2 \alpha_u^2}}, \\
k_{2,l} &= \frac{e^{-\frac{\beta_l^2}{2\left(\lambda_1 + \lambda_2 \alpha_l^2\right)}}}{\sqrt{\lambda_1 + \lambda_2 \alpha_l^2}},
\end{aligned}
\tag{D.21}
$$

we obtain

$$
\begin{aligned}
c_2 = & -\frac{\sqrt{\lambda_2}}{2\sqrt{2\pi}} \left[e^{-\frac{\tilde{x}_2^2}{2\lambda_2}} \operatorname{erf}\left(\frac{\alpha_u^2 \tilde{x}_2 + \beta_u}{\sqrt{2\lambda_1}} \right) - \operatorname{erf}\left(\frac{\alpha_l \tilde{x}_2 + \beta_l}{\sqrt{2\lambda_1}} \right) \right]_{\tilde{x}_{2,l}}^{\tilde{x}_{2,u}} \\
& + \frac{\lambda_2}{2\sqrt{2}} \left(\alpha_u k_{5,u} \left(\operatorname{erf}\left(\tilde{t}_u \right) - \operatorname{erf}\left(\tilde{t}_l \right) \right) - \alpha_l k_{2,l} \left(\operatorname{erf}\left(\tilde{s}_u \right) - \operatorname{erf}\left(\tilde{s}_l \right) \right) \right).
\end{aligned}
\tag{D.22}
$$

Calculating the integrals of c_1 and c_2 for every subregion, we obtain the conditional expectation in the transformed coordinate system which is then finally transformed into the original frame.

Bibliography

[1] P. L. Richardson, "Upwind dynamic soaring of albatrosses and UAVs," Progress in Oceanography, vol. 130, pp. 146–156, 2015.

[2] M. J. Allen and V. Lin, "Guidance and control of an autonomous soaring UAV with flight test results," 45th AIAA Aerospace Sciences and Meeting and Exhibit, January 2007.

[3] D. J. Edwards and L. M. Silberberg, "Autonomous Soaring: The Montague Cross-Country Challenge," Journal of Aircraft, vol. 47, pp. 1763–1769, 2010.

[4] H. Kon, "A numerical simulation of observed cumulus humilis clouds," Meteorological Society of Japan, Journal, vol. 58, pp. 262–272, 1980.

[5] J. Bolting, S. Fergani, J.-M. Biannic, F. Defaÿ, and M. Stolle, "Discrete sliding mode control of small UAS in tight formation flight under information constraints," IFAC-PapersOnLine, vol. 49, no. 17, pp. 332–337, 2016.

[6] S. Morton, R. D'Sa, and N. Papanikolopoulos, "Solar powered UAV: Design and experiments," in International Conference on Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ, pp. 2460–2466, IEEE, 2015.

[7] A. North, R. Siegwart, and W. Engel, "Autonomous solar UAV for sustainable flights," in Advances in Unmanned Aerial Vehicles, pp. 377–405, Springer, 2007.

[8] V. Bonnin, C. Toomer, J.-M. Moschetta, and E. Benard, "Energy Harvesting Mechanisms for UAV Flight by Dynamic Soaring," in AIAA Atmospheric Flight Mechanics (AFM) Conference, Guidance, Navigation, and Control and Co-located Conferences, (Boston, MA), August 2013.

[9] M. Deittert, A. Richards, C. A. Toomer, and A. Pipe, "Engineless unmanned aerial vehicle propulsion by dynamic soaring," Journal of Guidance, Control, and Dynamics, vol. 32, no. 5, pp. 1446–1457, 2009.

[10] J. W. Langelaan, "Long distance/duration trajectory optimization for small UAVs," in AIAA Guidance, Navigation and Control Conference and Exhibit, Hilton Head, SC, 2007.

[11] M. J. Cutler, T. W. McLain, R. W. Beard, and B. Capozzi, "Energy harvesting and mission effectiveness for small unmanned aircraft," in AIAA Guidance, Navigation, and Control Conference, Toronto, Canada, 2010.

[12] C. White, E. Lim, S. Watkins, A. Mohamed, and M. Thompson, "A feasibility study of micro air vehicles soaring tall buildings," Journal of Wind Engineering and Industrial Aerodynamics, vol. 103, pp. 41–49, 2012.

[13] J. Wharington, Autonomous control of soaring aircraft by reinforcement learning. PhD thesis, Royal Melbourne Institute of Technology, 1998.

[14] Michael J. Allen, "Autonomous soaring for improved endurance of a small uninhabited air vehicle," in 43rd AIAA Aerospace Sciences Meeting And Exhibit, January 2005.

[15] N. E. Kahveci, P. A. Ioannou, and M. D. Mirmirani, "Adaptive LQ control with anti-windup augmentation to optimize UAV performance in autonomous soaring applications," IEEE Transactions on Control Systems Technology, vol. 16, no. 4, pp. 691–707, 2008.

[16] J. Nguyen, N. Lawrance, R. Fitch, and S. Sukkarieh, "Energy-constrained motion planning for information gathering with autonomous aerial soaring," in IEEE International Conference on Robotics and Automation (ICRA), pp. 3825–3831, IEEE, 2013.

[17] M. Stolle, J. Bolting, C. Döll, and Y. Watanabe, "A vision-based flight guidance and navigation system for autonomous cross-country soaring UAVs," in Unmanned Aircraft Systems (ICUAS), 2015 International Conference on, pp. 109–117, IEEE, 2015.

[18] M. Stolle, Y. Watanabe, and C. Döll, "A sigma-point Kalman filter for remote sensing of updrafts in autonomous soaring," in Advances in Aerospace Guidance, Navigation and Control, pp. 283–302, Springer, 2015.

[19] J. A. Cobano, D. Alejo, S. Sukkarieh, G. Heredia, and A. Ollero, "Thermal detection and generation of collision-free trajectories for cooperative soaring UAVs," in 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2948–2954, IEEE, 2013.

[20] H. Reichmann, Cross-Country Soaring. Soaring Society of America, 1993.

[21] D. Pagen, Understanding the sky. Sport Aviation Publications, February 1992.

[22] J.-C. Golaz, H. Jiang, and W. R. Cotton, "A Large-Eddy Simulation study of cumulus clouds over land and sensitivity to soil moisture," Atmospheric Research, vol. 59, pp. 373–392, 2001.

[23] M. J. Allen, "Updraft model for development of autonomous soaring uninhabited air vehicles," in 44th AIAA Aerospace Sciences Meeting and Exhibit, American Institute for Aeronautics and Astronautics (AIAA), January 2006.

[24] N. R. Lawrance, Autonomous soaring flight for unmanned aerial vehicles. PhD thesis, University of Sydney, 2011.

[25] A. Renzaglia, C. Reymann, and S. Lacroix, "Monitoring the evolution of clouds with UAVs," in Robotics and Automation (ICRA), 2016 IEEE International Conference on, pp. 278–283, IEEE, 2016.

[26] S. C. Daugherty and J. W. Langelaan, Improving Autonomous Soaring via Energy State Estimation and Extremum Seeking Control. Pennsylvania State University, 2013.

[27] M. W. Hazard, "Unscented Kalman filtering for real-time atmospheric thermal tracking," Master's thesis, North Carolina State University, 2010.

[28] N. R. Lawrance and S. Sukkarieh, "Autonomous exploration of a wind field with a gliding aircraft," Journal of Guidance, Control, and Dynamics, vol. 34, no. 3, pp. 719–733, 2011.

[29] K. Andersson, I. Kaminer, and K. D. Jones, "Autonomous soaring: Flight test results of a thermal centering controller," in AIAA Guidance, Navigation and Control Conference, 2010.

[30] J. J. Acevedo, N. R. Lawrance, B. C. Arrue, S. Sukkarieh, and A. Ollero, "Persistent monitoring with a team of autonomous gliders using static soaring," in 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 4842–4848, IEEE, 2014.

[31] K. Andersson, I. Kaminer, K. D. Jones, V. Dobrokhodov, and D.-J. Lee, "Cooperating UAVs using thermal lift to extend endurance," in AIAA Unmanned Unlimited Conference, Seattle, WA, 2009.

[32] J. A. Cobano, D. Alejo, S. Vera, G. Heredia, S. Sukkarieh, and A. Ollero, "Distributed thermal identification and exploitation for multiple soaring UAVs," in Human Behavior Understanding in Networked Sensing, pp. 359–378, Springer, 2014.

[33] G. Halioui, "Conception d'algorithme de traitement d'image pour navigation autonome d'un drone planeur," Master's thesis, Ecole Nationale Supérieure de Physique de Strasbourg (ENSPS), 2016.

[34] C. W. Chow, B. Urquhart, M. Lave, A. Dominguez, J. Kleissl, J. Shields, and B. Washom, "Intra-hour forecasting with a total sky imager at the UC San Diego solar energy testbed," Solar Energy, vol. 85, no. 11, pp. 2881–2893, 2011.

[35] R. Johnson, T. Koehler, and J. Shields, "A multistation set of whole sky imagers and a preliminary assessment of the emerging data base," in Proceedings of the Cloud Impacts on DOD Operations and Systems, 1988 Workshop, pp. 159–162, 1988.

[36] T. Koehler, R. Johnson, and J. Shields, "Status of the whole sky imager database," in Proceedings of the Cloud Impacts on DOD Operations and Systems, 1991 Conference, pp. 77–80, 1991.

[37] A. Radovan and B. Željko, "Predictions of cloud movements and the sun cover duration," in 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), pp. 1210–1215, IEEE, 2014.

[38] S. Suzuki and K. Abe, "Topological structural analysis of digitized binary images by border following," Computer Vision, Graphics, and Image Processing, vol. 30, no. 1, pp. 32–46, 1985.

[39] S. S. Ponda, "Trajectory Optimization for Target Localization Using Small Unmanned Aerial Vehicles," Master's thesis, Massachusetts Institute of Technology, September 2008.

[40] Y. Watanabe, A. J. Calise, and E. N. Johnson, "Vision-based obstacle avoidance for UAVs," in AIAA Guidance, Navigation and Control Conference and Exhibit, vol. 11, 2007.

[41] V. J. Aidala, "Kalman filter behavior in bearings-only tracking applications," IEEE Transactions on Aerospace and Electronic Systems, no. 1, pp. 29–39, 1979.

[42] R. V. der Merwe and E. Wan, "The Unscented Kalman Filter for nonlinear estimation," in Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000, pp. 153–185, IEEE, October 2000.

[43] J. Civera, A. J. Davison, and J. M. Montiel, "Inverse depth parametrization for monocular SLAM," IEEE Transactions on Robotics, vol. 24, no. 5, pp. 932–945, 2008.

[44] K. Andersson and I. Kaminer, "Stability of a thermal centering controller," in AIAA Guidance, Navigation, and Control Conference,(Chicago, Illinois), AIAA, 2009.

[45] P. B. MacCready Jr., "Optimum airspeed selector," Soaring, pp. 10–11, 1958.

[46] R. Almgren and A. Tourin, "Optimal soaring via Hamilton–Jacobi–Bellman equations," Optimal Control Applications and Methods, vol. 36, no. 4, pp. 475–495, 2015.

[47] N. Kahveci, P. Ioannou, and D. Mirmirani, "Optimal static soaring of UAVs using vehicle routing with time windows," in AIAA Aerospace Sciences Meeting and Exhibit, AIAA-2007-158, AIAA, Washington, DC, USA, 2007.

[48] N. E. Kahveci, P. A. Ioannou, and M. D. Mirmirani, "A heuristic search algorithm for maneuvering of UAVs across dense thermal areas," in AIAA Guidance, Navigation and Control Conference and Exhibit, p. 6652, 2007.

[49] J. M. Wallace and P. V. Hobbs, Atmospheric Science: An Introductory Survey, vol. 92. Academic press, 2006.

[50] N. Kahveci, Robust Adaptive Control For Unmanned Aerial Vehicles. PhD thesis, University of Southern California - Faculty of the Graduate School, USA, 2007.

[51] M. G. Lawrence, "The relationship between relative humidity and the dewpoint temperature in moist air: A simple conversion and applications," Bulletin of the American Meteorological Society, vol. 86, no. 2, pp. 225–233, 2005.

[52] S. A. Ackerman and J. A. Knox, Meteorology: Understanding the Atmosphere. Cengage Learning, 2006.

[53] T. Bailey, J. Nieto, J. Guivant, M. Stevens, and E. Nebot, "Consistency of the EKF-SLAM algorithm," in Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on, pp. 3562–3568, IEEE, 2006.

[54] H. Shoudong and G. Dissanayake, "Convergence analysis for Extended Kalman Filter based SLAM," in Proc. of IEEE International Conference on Robotics and Automation. Orlando, USA, 2006.

[55] A. Huster, Relative position sensing by fusing monocular vision and inertial rate sensors. PhD thesis, Citeseer, 2003.

[56] S. Thrun, W. Burgards, and D. Fox, Probabilistic Robotics. Intelligent Robotics and Autonomous Agents series, The MIT Press, August 2005.

[57] S. Julier, "A skewed approach to filtering," in In SPIE Conference on Signal and Data Processing of Small Targets, vol. 3373, pp. 271–282, SPIE, April 1998.

[58] J. W. Langelaan, State estimation for autonomous flight in cluttered environments. PhD thesis, Stanford University, March 2006.

[59] R. V. der Merwe, "The square-root unscented Kalman filter for state and parameter estimation," in Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01). 2001 IEEE, vol. 6, pp. 3461–3464, May 2001.

[60] P. C. Mahalanobis, "On the generalized distance in statistics," Proceedings of the National Institute of Sciences (Calcutta), vol. 2, pp. 49–55, 1936.

[61] M. Montemerlo and S. Thrun, FastSLAM: A Scalable Method For The Simultaneous Localization and Mapping Problem in Robotics. Springer, 2007.

[62] M. Buhren and B. Yang, "Initialization procedure for radar target tracking without object movement constraints," in 2007 7th International Conference on ITS Telecommunications, pp. 1–6, IEEE, 2007.

[63] T. K. Yaakov Bar-Shalom, X. Rong Li, Estimation with Applications To Tracking and Navigation. John Wiley and Sons, Inc., 2001.

[64] R. W. Floyd, "Algorithm 97: shortest path," Communications of the ACM, vol. 5, no. 6, p. 345, 1962.

[65] J. W. Langelaan, "Tree-based trajectory planning to exploit atmospheric energy," in 2008 American Control Conference, pp. 2328–2333, IEEE, 2008.

[66] A. Chakrabarty and J. W. Langelaan, "Energy-based long-range path planning for soaring-capable unmanned aerial vehicles," Journal of Guidance, Control, and Dynamics, vol. 34, no. 4, pp. 1002–1015, 2011.

[67] T. Keviczky, F. Borrelli, K. Fregene, D. Godbole, and G. J. Balas, "Decentralized receding horizon control and coordination of autonomous vehicle formations," IEEE Transactions on Control Systems Technology, vol. 16, no. 1, pp. 19–33, 2008.

[68] T. Schouwenaars, Safe trajectory planning of autonomous vehicles. PhD thesis, Massachusetts Institute of Technology, 2005.

[69] R. Bellman, "A Markovian Decision Process," tech. rep., DTIC Document, 1957.

[70] R. A. Howard, Dynamic Programming and Markov Processes. The MIT Press, 1960.

[71] D. J. White, "A survey of applications of Markov Decision Processes," Journal of the Operational Research Society, vol. 44, no. 11, pp. 1073–1096, 1993.

[72] A. Y. Ng, A. Coates, M. Diel, V. Ganapathi, J. Schulte, B. Tse, E. Berger, and E. Liang, "Autonomous inverted helicopter flight via reinforcement learning," in Experimental Robotics IX, pp. 363–372, Springer, 2006.

[73] S. A. Quintero, G. E. Collins, and J. P. Hespanha, "Flocking with fixed-wing UAVs for distributed sensing: A stochastic optimal control approach," in 2013 American Control Conference, pp. 2025–2031, IEEE, 2013.

[74] S. S. Baek, H. Kwon, J. A. Yoder, and D. Pack, "Optimal path planning of a target-following fixed-wing UAV using sequential decision processes," in Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on, pp. 2955–2962, IEEE, 2013.

[75] W. H. Al-Sabban, L. F. Gonzalez, and R. N. Smith, "Wind-energy based path planning for unmanned aerial vehicles using Markov Decision Processes," in Robotics and Automation (ICRA), 2013 IEEE International Conference on, pp. 784–789, IEEE, 2013.

[76] A. Kolobov, "Planning with Markov Decision Processes: An AI Perspective," Synthesis Lectures on Artificial Intelligence and Machine Learning, vol. 6, no. 1, pp. 1–210, 2012.

[77] M. L. Puterman, Markov Decision Processes: Discrete Stochastic Dynamic Programming. John Wiley & Sons, February 2005.

[78] W. B. Powell, Approximate Dynamic Programming: Solving the Curses of Dimensionality. Wiley, 2 ed., September 2007.

[79] Z. Drezner and G. O. Wesolowsky, "On the computation of the bivariate normal integral," Journal of Statistical Computation and Simulation, vol. 35, no. 1-2, pp. 101–107, 1990.

[80] D. Müller and C. Kottmeier, Meteorologische Aspekte des Streckensegelfluges: Thermik von A bis Z. Müller & Kottmeier, 1986.

[81] C. Lindemann, "Some characteristics of thermal convection as measured by a powered sailplane," in Aero-Revue, 1978.

[82] D. Konovalov, "Thermals in the sub-cloud layer of the atmosphere," XV OSTIV Congress, 1976.

[83] R. Johnson, "Measurements of airplane sink rates between thermals," in OSTIV Publication, Aero-Revue, 1979.

[84] R. W. Beard and T. W. McLain, Small Unmanned Aircraft Theory and Practice. Princeton University Press, 2011.

[85] R. Brockhaus, W. Alles, and R. Luckner, Flugregelung. Springer, 3rd ed., 2011.