



Technische
Universität
Braunschweig



Deutsches Zentrum
DLR für Luft- und Raumfahrt

Master's Thesis

Development and testing of an exploration strategy for a cooperative network of rover and UAV using the example of mapping gas concentrations

Sebastian Bening

August 1, 2022



Deutsches Zentrum
DLR für Luft- und Raumfahrt

Prof. Dr.-Ing. Dirk Kügler

Supervisors:

Teemu Joonas Lieb, M. Sc.

Dr. Thomas Wiedemann

An die

Fakultät Maschinenbau
der Technischen Universität Braunschweig

Name Prof. Dr.-Ing. Dirk Kügler

Telefon +49 (0)531 295 2500

Telefax +49 (0)531 295 2550

E-Mail dirk.kuegler@dlr.de

31.01.2022

Thema der Masterarbeit für Herrn Sebastian Bening

Topic:

- Entwicklung und Erprobung einer Explorationsstrategie eines kooperativen Verbundes aus Rover und UAV am Beispiel der Kartierung von Gaskonzentrationen
- Development and testing of an exploration strategy for a cooperative network of rover and UAV using the example of mapping gas concentrations

Objective:

Technical progress enables the application of unmanned systems in new fields of activities. Amongst many other, unmanned vehicles are also used for the exploration of unsafe areas e.g. with potential hazardous gases. Hereby, the usage of networks of unmanned systems (like a cooperative network of a ground-based rover and a UAV) can increase the exploration speed as well as the information gained during the process while reducing the risks for human resources. Data gained by ground-based systems (here: by the rover) primarily provides information about the conditions near ground. At the same time, airborne systems (here: UAV) create turbulence and mix air masses which influences in-situ measurements. The innovative technique of "Tunable Diode Laser Absorption Spectroscopy" minimizes the influence of the airborne system by measuring the gas concentration along a laser beam.

The aim of this thesis is to develop a concept for the exploration and mapping of gas concentrations by using a cooperative network of unmanned vehicles both on the ground and in the air. The concept shall be based on a ground-based rover system equipped with a laser and an UAV (Unmanned Aircraft Vehicle) equipped with the receiving sensor. By measuring the absorption of the laser beam, the composition of the surrounding air can be determined. The main focus of this thesis is the development of an algorithm for the movement and cooperation of ground-based and airborne systems. The algorithm shall determine optimal positions for the unmanned vehicles in order to obtain efficient measurements of the gas concentrations. The cooperation with airborne systems alters the boundary conditions compared to mere ground-based systems. The boundary conditions for the concept shall be elaborated and quantified. Furthermore, the developed algorithm shall be implemented into a software application for the usage in simulations and demonstrations. In addition, software required for testing and verification purposes of the algorithm shall be chosen. Thereafter, the concept shall be validated in software simulations. The results shall be gathered and evaluated. Further, a system

capable to realize the exploration strategy shall be specified. Conclusive, a proof of concept demonstration via flight tests shall be performed and documented.

Work to be performed:

- Literature research of strategies for combined airborne and ground-based exploration
- Development of an exploration concept
- Development of an algorithm for the movement of the cooperating systems
- Development of software modules for air-ground cooperation and exploration
- Planning and execution of simulations of the exploration concept
- Validation of the exploration concept e.g. in software simulations
- Proof of concept demonstration for the exploration concept

Literature:

- [1] Sujit, P.B. and Beard R.
Cooperative Path Planning for Multiple UAVs Exploring an Unknown Region
2007 American Control Conference, July 2017, New York City, USA
- [2] Ropero, F; P. Muñoz, P. and R-Morena, M. D.
TERRA: A Path Planning Algorithm for Cooperative UGV-UAV Exploration
Engineering Applications of Artificial Intelligence, Volume 78, February 2019
- [3] Wiedemann, T.; Shutin, D. and Lilienthal, A. J.
Model-based gas source localization strategy for a cooperative multi-robot system — A probabilistic approach and experimental validation incorporating physical knowledge and model uncertainties
Robotics and Autonomous Systems, Volume 118, 2019, Pages 66-79

Zeitraumen:

Beginn: 31.01.2022

Dauer: 6 Monate

Betreuer:

Teemu Joonas Lieb
Institut für Flugführung / Abteilung PAS
Deutsches Zentrum für Luft- und Raumfahrt e.V.
Lilienthalplatz 7
38108 Braunschweig

Erstprüfer:

Professor Dr.-Ing. D. Kügler
Deutsches Zentrum für Luft- und Raumfahrt e.V.
Institut für Flugführung
Lilienthalplatz 7
38108 Braunschweig

Zweitprüfer:

Professor Dr.-Ing. P. Hecker
Technische Universität Braunschweig
Institut für Flugführung
Hermann-Blenk-Straße 27
38108 Braunschweig

Dirk Kuegler
Digital signiert von Dirk Kuegler
DN: C=DE, O=Deutsches Zentrum
für Luft- und Raumfahrt e.V.
(DLR), CN=Dirk Kuegler
Grund: Ich genehmige dieses
Dokument
Ort: Wolfenbüttel
Datum: 2022.01.31
21:28:43+01'00'
Foxit PDF Editor Version: 11.1.0

Prof. Dr.-Ing. D. Kügler

Statement of Originality

This thesis has been performed independently with the support of my supervisor/s. To the best of the author's knowledge, this thesis contains no material previously published or written by another person except where due reference is made in the text.

Braunschweig, August 1, 2022

S. Bening

Abstract

The present thesis develops and demonstrates an exploration strategy for the reconnaissance of gas plumes with a cooperative network of airborne and ground-based unmanned vehicles by remote sensing. The cooperation of airborne and ground-based vehicles poses a clear distinction from other work in this field. For the measurement of the gas plume, the method of Tunable Diode Laser Absorption Spectroscopy (TDLAS) is applied. The method of TDLAS requires a multitude of measurements from different angles. Using the airborne vehicle as a reflector for the TDLAS beam provides more precise information on the measurement. To optimise the positioning of the vehicles and explore the gas plume, an engineering approach is pursued. A heuristic containing an exploration, an exploitation, and a moving costs term is developed. Furthermore, a system capable of implementing the strategy is arranged and presented. In a multi-step approach, the exploration concept is implemented and verified. The thesis presents the structure of the implementation of the exploration strategy in a Python simulation. With the simulation, the basic operating principle of the exploration strategy has been verified. Supplementary, the parameters used in the heuristic are examined to improve the performance of the exploration strategy. Besides, different variations of the strategy are analysed and discussed. Additionally, the implementation is transferred to a physics simulation with Robot Operating System (ROS) and Gazebo. In the simulation, the strategy is combined with the constraints of the vehicles. The simulation is performed with software in the loop simulations of the actual hardware to verify the compatibility. In a proof-of-concept demonstration, the implementation of the exploration strategy in the real hardware system (i.e. Unmanned Aerial Vehicle (UAV) and Unmanned Ground Vehicle (UGV)) with software-in-the-loop simulation of the TDLAS sensor is shown and analysed.

Zusammenfassung

In der vorliegenden Masterarbeit wird eine Strategie zur Erkundung von Gaswolken mittels unbemannter Luft- und Bodenfahrzeuge entwickelt. Die Vermessung der Gaskonzentration beruht auf dem Verfahren der Tunable Diode Laser Absorption Spectroscopy (TDLAS). Dieses Verfahren benötigt Messungen aus verschiedenen Winkeln, um eine Rekonstruktion der Gaswolke zu ermöglichen. Die Verwendung des Luftfahrzeugs als Träger der Reflektionsfläche ermöglicht die Durchführung von reproduzierbaren Messungen. Zur Optimierung der Fahrzeugposition und der Erkundung der Gaswolke wird ein ingenieurmäßiges Vorgehen verfolgt. Das bedeutet, es wird eine Heuristik aufgestellt, die einen untersuchenden und einen erschließenden Anteil hat, sowie den Aufwand für die Bewegung in Betracht zieht. Ergänzend wird ein System dargestellt, welches in der Lage ist, die Erkundungsstrategie zu realisieren. Alle benötigten Hardware- und Softwarekomponenten werden erläutert. In einem mehrstufigen Ansatz wird das entwickelte Konzept implementiert und verifiziert. Im ersten Schritt wird eine Pythonsimulation aufgesetzt, um die grundsätzliche Funktionsweise des Konzepts zu verifizieren und analysieren. Anschließend wird diese Simulation verwendet, um die Parameter der Heuristik zu bestimmen. Zusätzlich werden mit Hilfe der Simulation verschiedene Variationen der Strategie untersucht und anhand der Ergebnisse diskutiert. Im nächsten Schritt wird die Implementierung in eine physikalische Simulation mit dem Robot Operating System (ROS) und Gazebo überführt. Mit dieser zweiten Simulation wird die Funktionsweise des Systems mit den Einschränkungen der Hardware verifiziert. Weiterhin wird die Simulation genutzt, um die Software zur Umsetzung der Strategie mit den Fahrzeugen zu erproben. Im dritten und letzten Schritt wird die Strategie in einer Machbarkeitsstudie mit existierenden Fahrzeugen demonstriert und die erzielten Ergebnisse diskutiert. Am Ende der vorliegenden Arbeit wird ein Ausblick zu weiterer Forschung im Bereich der kooperativen unbemannten Fernerkundung von Gaswolken gegeben. Hierbei werden zum einen Verbesserungen des entwickelten Konzepts, als auch Abwandlungen, vorgeschlagen und erläutert.

Contents

List of Acronyms	1
1. Introduction	2
1.1. Motivation	2
1.2. Objective	3
1.3. Contribution and Outline	4
2. Theoretical Principles	5
2.1. Remote Sensing	5
2.1.1. Overview	5
2.1.2. Tunable Diode Laser Absorption Spectroscopy	6
2.1.3. Unmanned Gas Exploration	7
2.2. Applied Algorithms	8
2.2.1. Path Creation	8
2.2.2. Bresenham Line Algorithm	10
3. Exploration Concept	13
3.1. Problem Statement and Analysis	13
3.2. Gas Distribution Mapping by Remote Sensing	14
3.3. Exploration Strategy	17
3.3.1. System Concept	17
3.3.2. Measuring Point Selection	19
3.3.3. Patterns	21
3.3.4. Communication	23
3.3.5. Extension Capabilities	23
4. Implementation	25
4.1. System Setup	25
4.1.1. Airborne System	25
4.1.2. Ground System	26
4.1.3. Sensor System	27
4.1.4. ROS	29
4.2. Simulation	32
4.2.1. Assumptions	32
4.2.2. Coordinate Systems	32
4.2.3. Scenarios	34
4.2.4. Evaluation Criteria	37

4.2.5. Python Simulation	38
4.2.6. Physics Simulation	41
4.2.7. Roadmap for the Simulation Runs	44
4.3. Field Tests	45
5. Results	48
5.1. Python Simulation Results	48
5.1.1. Regularisation Factor	48
5.1.2. Random Approach	50
5.1.3. Cost Function Parameters	53
5.1.4. Pattern of Interest	59
5.1.5. Best Beam	69
5.1.6. Comparison	71
5.1.7. Plume Reconstruction	72
5.2. Physics Simulation Results	79
5.3. Field Test Realisation and Results	82
6. Summary	85
7. Conclusion	87
Bibliography	88
A. Project management	93
A.1. Work Breakdown Structure	93
A.2. Timetable	95
B. Storage Device	98

List of Figures

2.1.	Working principle of a TDLAS sensor	6
2.2.	Path finding process with the A* algorithm	9
2.3.	Visualisation of the travelling salesperson problem	10
2.4.	Principle of the Bresenham line algorithm	11
3.1.	Considered spaces	14
3.2.	Two-dimensional example for a TDLAS measurement	15
3.3.	Visualisation of the process of multiple measurements	16
3.4.	Measuring variants	18
3.5.	Visualisation of the patterns	22
4.1.	Holybro S500 used for the demonstration	26
4.2.	Summit XL used for the demonstration	27
4.3.	TDLAS sensors	28
4.4.	Gimbals for mounting the sensor	29
4.5.	Principle of a ROS network	30
4.6.	Complex ROS network	31
4.7.	Map generation	33
4.8.	Voxel enumeration and coordinate systems	34
4.9.	Environments	35
4.10.	Plumes	37
4.11.	Process of the Python simulation	39
4.12.	Physics Simulation	42
4.13.	Setup hardware tests	47
5.1.	Plume error for different regularisation factors.	48
5.2.	Condition number for the plume reconstruction	49
5.3.	Analysis of random measurements with the centred plume	51
5.4.	Analysis of random measurements in Mo-o	52
5.5.	Plume error for evaluation of d	53
5.6.	Distance for the evaluation of d	54
5.7.	Plume error for the evaluation of c	56
5.8.	Plume error for m evaluation with hourglass pattern	57
5.9.	Substantiating plots for the evaluation of m	58
5.10.	Evaluation of v	59
5.11.	Evaluation results for the hourglass pattern	61
5.12.	Evaluation results for the cross pattern	62

5.13. Evaluation results for the inverted pyramid pattern	63
5.14. Evaluation results for the pyramid pattern	65
5.15. Evaluation results for the cube pattern	66
5.16. Plume error for the pattern comparison	67
5.17. Pattern comparison distance travelled by the vehicles	68
5.18. Results of the best beam evaluation	70
5.19. Evaluation of all target allocation methods	71
5.20. Distance travelled for all measuring methods	72
5.21. Hourglass_9_4_4 plume reconstruction in the Mo-o scenario with the cen- tered plume after 100 beams	73
5.22. Hourglass_9_4_4 plume reconstruction in the Mo-o scenario with the cen- tered plume after 500 beams	74
5.23. Hourglass_9_4_4 plume reconstruction in the Mo-o scenario with the cen- tered plume after 1000 beams	75
5.24. Hourglass_9_4_4 plume reconstruction in the M2-2 scenario with the lay- ered double plume after 100 beams	76
5.25. Hourglass_9_4_4 plume reconstruction in the M2-2 scenario with the lay- ered double plume after 500 beams	77
5.26. Hourglass_9_4_4 plume reconstruction in the M2-2 scenario with the lay- ered double plume after 1000 beams	78
5.27. Hourglass_9_4_4 plume reconstruction in the M2-2 scenario with the lay- ered double plume after 6000 beams	79
5.28. Screenshot of the physics simulation	80
5.29. Evaluation of the plume error in the physics simulation	81
5.30. Documentation of the field tests	82
5.31. Sections of vehicle paths during proof-of-concept demonstration	84

List of Acronyms

CBRN Chemical, Biological, Radiological, Nuclear

DLOS Direct Line of Sight

DLR German Aerospace Center

GNSS Global Navigation Satellite System

GPS Global Positioning System

Lidar Light Detection and Ranging

NRMSE Normalised Root Mean Square Error

Radar Radio Detection and Ranging

RMSE Root Mean Square Error

ROS Robot Operating System

RTK Real Time Kinematic

TDLAS Tunable Diode Laser Absorption Spectroscopy

UAV Unmanned Aerial Vehicle

UGV Unmanned Ground Vehicle

WLAN Wireless Local Area Network

1. Introduction

1.1. Motivation

Humankind always strived to discover the unknown and exemplifies a spirit of discovery. While the exploration of Earth was performed by humans themselves for a long period of time, expanding the exploration to space led to the development of unmanned exploration. In the 1950s, when humankind started grasping for the stars, the first objects in space were unmanned vehicles [34] [55]. Afterwards, the first objects sent to the Moon, Mars, or even the outer solar system were unmanned [56] [22] [32]. Until today, the farthest distance a human travelled was to the far side of the Moon [52]. Whereas the farthest unmanned vehicles Voyager 1 is in interstellar space at a distance of more than 23 billion kilometres [32]. On the surface of Mars, rovers are exploring the red planet with increased autonomy required due to the high communication delay of up to 20 minutes [30]. In the near past, the first cooperative exploration group consisting of the rover Perseverance and the so-called Mars-helicopter Ingenuity started exploring Mars [31]. The combined capabilities of the vehicles increase the quality of the gained data and improve the acquisition speed itself. Promising locations can be identified by the airborne vehicle and investigated in detail by the ground vehicle.

However, with the recent technological advance, unmanned exploration on Earth is becoming more and more viable. The knowledge gained in space exploration can be transferred to areas and use cases down on Earth. The initial intention of using unmanned systems was the exploration without endangering human life. Unmanned systems can be used to explore potentially dangerous areas to reduce the risk for human operators [26]. This can be for example areas with concrete danger of falling objects or near buildings at risk of collapsing after an earthquake or another natural disaster. Though, it can also be areas where toxic gases are expected for example after volcanic activity or after an accident in an industrial processing site. The potential of using Unmanned Aerial Vehicle (UAV)s for monitoring Chemical, Biological, Radiological, Nuclear (CBRN) threats in industrial areas is investigated in [42]. Unmanned systems can be used to locate the gas sources [54]. In these cases, unmanned systems can be used to gain information about the situation to safeguard human life [46]. To go one step further, autonomous operating vehicles can be used to constantly monitor areas with an increased risk for hazards [41]. Whilst regular inspections of large industrial sites require a great amount of manpower and thereby high expenses, unmanned systems can continuously patrol such facilities and alert human operators in case of abnormalities. Based on the information provided by the unmanned systems, the operators can act. This leads to a faster and improved reaction with a reduction of costs at the same time.

As an example, today landfills are a common way to dispose of huge amounts of waste. During the decomposition of waste, toxic gases like methane [3] arise. High concentrations of methane would be harmful to the climate whilst its flammability and explosiveness combined with ambient air pose a serious danger to human operators. For this reasons, it is desirable to detect methane leakages early to resolve the leakage. Due to the number of landfills and their size, it is not possible to fully cover these tasks with human operators. A similar situation can be found in refineries or natural gas and oil processing sites in general. A large proportion of the natural gas is methane. The leakages lead to an increased contribution to man-made climate change [28].

In the present thesis, a concept is developed to use a cooperative group of airborne and ground-based unmanned vehicles for continuous monitoring and mapping of gas plumes to counteract these issues and reduce risks for human operators. For the detection of gas in three-dimensional space, the method of Tunable Diode Laser Absorption Spectroscopy (TDLAS) is applied. The development of the thesis is supervised by the German Aerospace Center (DLR) Institutes of *Flight Guidance* and *Navigation and Control*.

1.2. Objective

The main objective of the present thesis is the development of an exploration strategy for the autonomous mapping of gas plumes by cooperating unmanned systems. The concept can be used in hazardous environments to reduce the risk for human operators and simultaneously improves the efficiency of the exploration procedure by using autonomous vehicles. Possible use cases are the detection and mapping of gas concentrations in natural gas processing facilities or above landfills. Concentrations of climate-damaging or dangerous gases for humans or rather living beings in general should be monitored continuously in a safe manner. For this reason, fast detection of the gas concentrations is required. Compared to other research in the area of unmanned exploration, this thesis discusses an exploration concept to enable autonomous exploration by remote sensing with TDLAS by a cooperation of unmanned airborne and ground-based systems. The TDLAS is used to determine the gas distribution in the air. A two-part system, half ground-based, half airborne reduces the influence on the gas distribution by turbulence and mixing caused by the system itself. An in-situ sensor mounted to a UAV would influence the gas volume by its downwash and distort the measurement. At the same time, the two-part system enables measuring and extending areas that could not have been measured only with a ground-based in-situ sensor. This thesis focuses on the development of an algorithm for the movement planning for a UAV and an Unmanned Ground Vehicle (UGV) through their environment to increase the knowledge gained about gas distribution by improving the significance of measurements. Based on an engineered heuristic, the algorithm determines a position for the vehicles which maximises the information gained with the next measurements of the TDLAS-sensor. Therefore, the system also takes its knowledge about the surroundings into account.

1.3. Contribution and Outline

The present master's thesis is introduced with the presentation of a set of theoretical principles (Chapter 2). Those were studied and chosen as a foundation for the developed concept. The section leads with a short overview of remote sensing in general then followed by a presentation of the tunable diode laser absorption spectroscopy in Section 2.1.2. Afterwards, approaches for gas exploration with unmanned vehicles are presented in Section 2.1.3. This subsection situates the present master's thesis in the current scientific state of the art. The chapter is concluded with a section on selected algorithms (Section 2.2) applied during this thesis. These existing path planning algorithms are used to define the path considering the limitations of the unmanned systems required for the developed concept. In Chapter 3 the developed exploration concept is explained. To the best knowledge of the author, it is the first time that it is investigated to combine an unmanned ground vehicle with an unmanned airborne vehicle to measure gas plumes in three-dimensional space with the TDLAS principle. In the beginning, the problem the thesis is based on is analysed. In the next subsection, a solution is derived from the results of the analysis on which this thesis is based. Concluding the chapter ends with the exploration strategy developed in this thesis (Section 3.3). In the following Chapter 4, the multi-step implementation chosen for the approach of the thesis is presented. The chapter is introduced with the presentation of hardware and software required to realise the developed exploration concept. Following, the two simulations developed in the course of this thesis are described. In the first step, the characteristics of the Python-based simulation and its main procedure is outlined. In the second step, the concept is implemented in a Robot Operating System (ROS) [45] system to control and monitor the unmanned systems required for the realisation of the concept. The software is evaluated in a physics simulation performed with Gazebo [17]. Once the exploration strategy is verified within the simulation, a proof-of-concept demonstration is undertaken in form of field tests with unmanned systems. The design for the proof-of-concept tests is presented at the end of the chapter. For the present thesis, numerous simulation runs have been conducted. A detailed discussion of the simulation results is set out in Chapter 5. The chapter covers the Python simulation as well as the physics simulation performed with Gazebo. Conclusively, the execution of the field test performed in the scope of the present thesis is documented and the findings are discussed. The thesis is finalised with a summary of the achieved results and findings in Chapter 6. Closing, a conclusion, as well as an outlook for further work, is given in Chapter 7.

2. Theoretical Principles

This section is used to provide the reader with a basic knowledge of theoretical principles applied during the development of this thesis and the implementation of the presented concepts. It leads with an introduction of remote sensing and more specified on the topic of TDLAS. Afterwards, an overview of research in the area of unmanned gas exploration is provided. At the end, a short presentation on the basics of algorithms applied in this thesis is provided.

2.1. Remote Sensing

2.1.1. Overview

Remote sensing is a method to examine the physical characteristics of an area from the distance. Emitted and reflected radiation is detected and monitored from a certain distance. Usually, the measurements are performed with an aircraft or even satellites [33]. In contrast to in-situ measurements, remote sensing is able to measure the value of interest over a large area in a short amount of time. Though, in-situ measurements provide more precise information about the local conditions and a higher positional resolution of the measured values. Remote sensing sensors can either be active or passive. Passive sensors rely on emitted or reflected natural energy, most of the time sunlight. Such are radiometers and spectrometers. Passive sensors often are limited by cloud cover and cannot penetrate dense clouds [33]. Active sensors emit the measured radiation themselves. It includes different Radio Detection and Ranging (Radar) and Light Detection and Ranging (Lidar) sensors. The sensors mostly operate in the microwave band which enables them to penetrate the atmosphere most of the time [33]. Remote sensing has a wide field of applications. It ranges from environmental monitoring over hazard assessment up to natural resource management [37]. Remote sensing has a series of advantages. As mentioned before, remote sensing can easily be used to cover large areas. In addition, it is possible to use it for constant or at least repetitive coverage of an area. Thereby, the temporal course can be monitored closely [2]. Furthermore, remote sensing does not take place directly in the area of interest and consequently does not disturb the measured object. On the other hand, it also comes with some disadvantages. If applied to small areas, it results in rather high costs which may be prevented by in-situ measurements [2]. Remote sensing data is more complex to be analysed than in-situ measurements and therefore special trained personnel and processing equipment is required [2]. This may result in higher overall costs depending on the use case. In the close past, also the integration of remote sensing equipment onto smaller unmanned vehicles can be observed. This topic

is discussed further in Section 2.1.3.

2.1.2. Tunable Diode Laser Absorption Spectroscopy

A remote sensing method using an active sensor is the so-called Tunable Diode Laser Absorption Spectroscopy (TDLAS). Further on in this work, the presented concept is designed for measuring gas concentrations and shall therefore be introduced at this point.

The method of TDLAS is often used for gas detection because it is a fast process providing a high resolution and sensitivity [18]. The technique of TDLAS exploits the Beer-Lambert law [49]. Radiation passing through an absorbing medium is dampened depending on the concentration and layer thickness of the plume. The Beer-Lambert law states a logarithmic relation between the transmission of light through a gas, the attenuation coefficient of the gas, and the distance light travels through the gas [29]. In Fig. 2.1 the measurement process applied in this thesis is depicted. In order to measure, a beam is created by a tunable diode laser. Tunable diode lasers are used because it is possible to tune the wavelength fast and to achieve fast response times [53]. Inherent stability and single frequency emission are important to achieve narrow line widths and thereby high-quality measurements [53]. The beam passes through the gas until it hits a solid object that reflects it back. After the beam passes through the gas a second time, it hits the photodetector next to the laser emitter. In stationary systems, the photodetector is directly at the other side of the gas and the beam crosses the gas only once. Not only does the dampening by the gas influence the measurement, but also the characteristics of the reflecting material. With a modification of the wavelength, different gases can be detected by the beam.

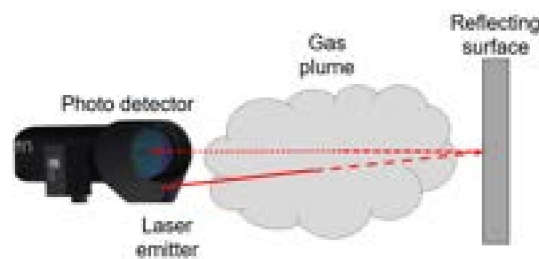


Figure 2.1.: The figure shows the working principle of a TDLAS sensor. A laser source emits a beam that radiates through the gas. On the other side of the plume, it is reflected on a surface. A photodetector above the laser source collects the reflected light. During the transition through the gas, the laser beam is dampened in correlation to the concentration and thickness of the gas (Sensor image source [38]).

The result provides an integrated measurement along the path of the laser beam. With a single measurement, it can only be determined that a specific gas is inside the area. No statement can be made about the spatial distribution. To receive this information, multiple measurements from different directions have to be performed. For calculating the spatial distribution from the measurements, a tomographic reconstruction has to be

performed. The procedure is similar to medical computer tomography. In that case, the beams are x-rays. The examined objects are solid and not gaseous. With technological advances, the size of TDLAS sensors has been reduced so that they could be transformed into a mobile format. This enables in-field usage and live measurements. In [21] a static TDLAS system was installed above a landfill in China and multiple measurements have been performed to learn more about the methane outgassing in the area. These aspects are being used during the concept discussed in this thesis.

2.1.3. Unmanned Gas Exploration

In the current research, multiple approaches for the exploration of gas concentrations in an area are pursued. This is often motivated to detect climate-damaging gases at an early stage and prevent further damage. In many cases, unmanned vehicles are utilised in these approaches to achieve autonomy and improve the efficiency of the process.

In [54], a swarm of unmanned ground vehicles with in-situ sensors is used to obtain knowledge on the gas dispersion in an area. In the paper, it is investigated whether the measuring process and with it the localisation of the gas source can be improved. Therefore, it is striven to include the gas dispersion process. The experiments performed in [54] substantiate the initial claim. Even though this approach shows a fast exploration of an area, it is limited to exploration close to the ground in an area due to the limits emerging from the ground-based in-situ sensors.

By contrast, in [48] a single agent system with multiple sensors for remote sensing is introduced. The presented system combines a passive infrared thermographic imaging sensor as well as an TDLAS sensor with a ground-based vehicle called RoboGas Inspector [48]. This combination allows the system to measure gas concentration and identify leakages above ground level. It is designed to be used for inspection of larger industrial sites [48]. The application of unmanned systems simplifies inspecting remote areas and establishes a sufficient solution for monotonous and repetitive tasks [48]. In addition, it allows for more efficient deployment of the specialised staff [48].

In [6] another ground-based unmanned system using the TDLAS method is introduced. The paper concentrates on developing an optimised exploration strategy for robot-assisted gas tomography. Therefore, the developed algorithm distinguishes between local high concentration regions and the overall global solution [6]. In the process at first promising measuring geometries have been identified which then were integrated into an optimisation framework for larger areas. In order to verify the concept, it is demonstrated in simple simulations as well as in a real-world experiment [6]. In contrast to the present thesis, the concept and test presented in [6] are developed for the two-dimensional case.

In comparison, in [35] and [36] an airborne system carrying a TDLAS sensor is presented. In this case, an autonomous aerial robot is designed to reconstruct tomographic two-dimensional slices of gas plumes in outdoor environments [35]. The system is based on an octocopter with a three-axis gimbal. With its maximum take-off weight, it can reach flight

times up to 15 minutes [35]. The positioning of the UAV is based on a Global Navigation Satellite System (GNSS) solution [35]. A TDLAS sensor is used for the measurement of gas concentrations and complemented by a laser range finder to calculate the length of the beam [36]. For calculating the model of spatial gas distribution, computer tomography with a grid-based approach is applied [35]. The verifications performed in an outdoor environment showed that the sensor requires a certain intensity of reflection to work, depending on the surface of the reflecting material [35].

As it is outlined in this section, multiple approaches for the unmanned exploration of gas distributions have been researched in the past. None of these combines a ground-based vehicle carrying a TDLAS sensor with an airborne vehicle to achieve a defined reflection behaviour. With this combination, the present thesis introduces a new approach to measuring three-dimensional gas concentrations.

2.2. Applied Algorithms

The present thesis utilises existent algorithms for solving objectives of different manners. In this section, a short introduction to these algorithms is outlined.

2.2.1. Path Creation

In this thesis, two path creation tasks are performed. For each vehicle, a list of target waypoints is created with points that shall be visited in the next measuring cycle. In a first step, the path between all pairs of waypoints is determined. The path between two waypoints of a vehicle shall be as short as possible without colliding with any obstacles in the area. Therefore, a version of the A*-Algorithm is used. It is a common algorithm for path planning in known environments. The second task is to arrange all the points that shall be visited to obtain a minimal movement time for the vehicles between the waypoints. This part is solved similarly to the Travelling Salesperson Problem. Following an introduction to these algorithms is provided.

The A* Algorithm

Path planners are used to find the best path from a given starting point to the targeted location. Vehicles relying on short-range sensors can only react to obstacles in their path. Especially U-shaped or V-shaped obstacles can lead to an increased travelling distance. Path planners can be used to avoid getting caught by these obstacles. Path planners analysing larger areas can be used to detect such obstacles and plan a path that initially appears longer, but bypasses obstacles and thereby leads to a shorter overall distance [40].

The so-called A* algorithm is an informed search algorithm. In opposite to uninformed algorithms, it only works if the environment is known, wherein the movement is performed. The A* algorithm uses heuristics to find the optimal path to the target. It is

based on Dijkstra's Algorithm [40]. The Dijkstra Algorithm is an algorithm to find the shortest path between nodes of a graph [12]. It expands its search outward beginning at the starting point. It does so by checking every closest point that has not been visited yet [40]. This behaviour is similar to the greedy-best-first algorithm. The algorithm calculates the distance between the possible points and the goal. Then that algorithm selects the point that is closest to the target. The greedy-best-first algorithm has improved performance compared to the Dijkstra algorithm but then it is not guaranteed that it finds the shortest solution [40]. The A* algorithm combines both approaches and takes the distance from the start as well as the distance towards the goal into account. Its heuristic can be formulated as follows [40]:

$$f_n = g_n + h_n \quad (2.1)$$

The total cost f_n for the next point n is the sum of the distance from the starting node to the point g_n and the estimated distance from the point to the target h_n . In Fig. 2.2 an example is depicted. The total cost is calculated for all possible next points until the target is reached. For the heuristic multiple variants to calculate the distance to the target are possible. One possibility is the so-called Manhattan distance that does not allow diagonal movement as shown in Fig. 2.2. Movements are only allowed along the sides of the squares. This results in [39]:

$$h_n = |x_n - x_{target}| + |y_n - y_{target}| \quad (2.2)$$

A different approach would allow diagonal movement by applying the euclidean distance. In this case, the points 2, 5, and 9 would not be visited. The heuristic would then be calculated with [39]:

$$h_n = \sqrt{(x_n - x_{target})^2 + (y_n - y_{target})^2} \quad (2.3)$$

4	3	4	5	6	7	8	9	10	11
3	2	3	4				10	11	12
2	1	2	3	4		12	11	12	13
1	0	1	2	3		13	12	13	14
2	1	2	3	4		12	13	14	15
3	2	3	4	5		11	12	13	14
4	3	4	5	6		10	11	12	13
5	4	5	6	7	8	9	10	11	12

Figure 2.2.: The figure shows the path-finding process applied with the A* algorithm. The starting point is in the dark green square 0. The target point is in the blue square 13. The vehicle cannot pass through the grey marked cells because these represent obstacles. The solution based on the A* algorithm is the light green path.

The Travelling Salesperson Problem

The travelling salesperson problem addresses the difficulty of connecting a group of points with the shortest tour possible. Thereby, it is a typical optimisation problem [20]. In Fig. 2.3 the problem is depicted. A vehicle located in the blue marked position zero has the task to visit the six marked positions in a single tour. The problem can either include the return to the home position or stop at the last point. In this case, the vehicle does not have to return to its origin. The goal is to visit each of the six points once and choose the shortest possible tour. The travelling salesperson problem cannot be solved in polynomial time, which means that the time to acquire a solution increases exponentially with the number of visited points [4]. Nevertheless, for short lists of points, as it is the case in the present thesis, it is still viable to calculate the exact solutions [4]. More detailed information on the use in this thesis is provided in Section 4.2.5. For longer list the randomized optimisation algorithm as described in [20] is potentially a better approach. A different approach for complex problems would again be the use of heuristics. In [4] different heuristics are presented. At this point no deeper discussion of other possible solutions will be performed, since this thesis applies an already implemented travelling salesperson solver as described in [51] and does not attempt to optimise the solver.

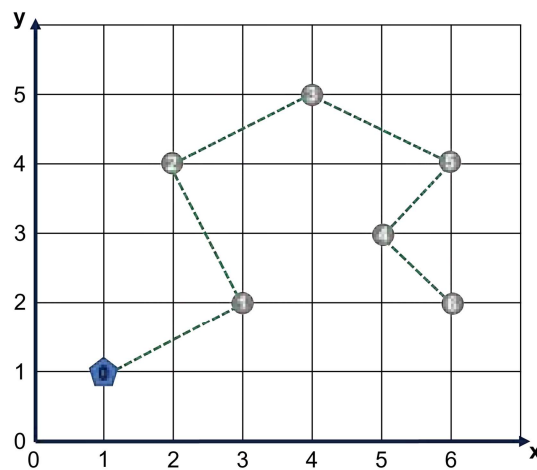


Figure 2.3.: The travelling salesperson problem is trying to find the shortest tour between multiple points. The vehicle is at starting point 0 and wants to visit every point on the route exactly once.

2.2.2. Bresenham Line Algorithm

The Bresenham line algorithm was developed by Jack Elton Bresenham in 1962 for the computer control of a digital plotter [9]. The line drawing algorithm can be used to approximate a straight line between two points in an n -dimensional raster. The algorithm exists since the beginning of the computer age and is known due to its simplicity and the use of resource inexpensive operations [9].

The coordinates of the start point are $P_1 = (x_1, y_1)^T$ and for the target point $P_2 = (x_2, y_2)^T$. It is established that the x-coordinate increases towards the right and the y-coordinate towards the top. Furthermore, it must apply that the crossings of the grid have integer coordinates. The procedure and conventions are visualized in Fig. 2.4.

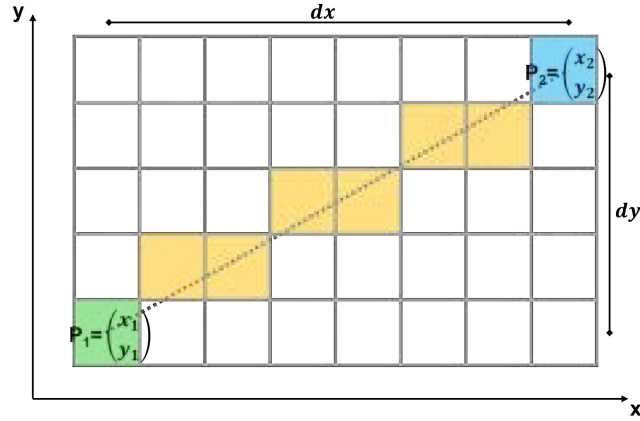


Figure 2.4.: The figure shows the operating principle of the Bresenham line algorithm in a two-dimensional grid. The algorithm is used to find the direct connection between P_1 and P_2 within the grid. For each step in the x-direction, exactly one voxel is taken into account.

At the beginning it has to be evaluated in which direction the distance between the points is the greatest by defining $\Delta x = x_2 - x_1$ and $\Delta y = y_2 - y_1$. In the shown case, it results in $\Delta x > \Delta y$. Therefore, the x-axis is the driving axis. In every following step, it moved forward along the x-axis and sometimes along the y-axis. A slope error variable d is used to identify the necessity for steps along the y-axis. Initially, the slope error is defined as:

$$d = 2 \cdot \Delta y - \Delta x \quad (2.4)$$

Further, it is defined for the error correction:

$$d_O = 2 \cdot \Delta y \quad (2.5)$$

$$d_{NO} = 2 \cdot (\Delta y - \Delta x) \quad (2.6)$$

After the initial definitions, the algorithm increases the driving coordinate in each step by one while $x < x_2$.

$$x = x + 1 \quad (2.7)$$

At the same time, the size of the error d is evaluated. If it is smaller than zero, then d_O is added:

$$d = d + d_O \quad (2.8)$$

If the error becomes larger than zero, the error is corrected by d_{NO} and the value of the y -axis is increased by one:

$$d = d + d_{NO} \quad (2.9)$$

$$y = y + 1 \quad (2.10)$$

This procedure is repeated until x_2 is reached. The algorithm can be modified to be applicable in the three-dimensional space. Then it is not used for determining pixels but voxels. To expand it to the third dimension, the same process as shown above is repeated for the third dimension. Therefore, an additional error variable is introduced. It correlates with the third axis. If it exceeds zero, the value along the third axis is increased. This extension is required because the beam of the TDLAS sensor in the present thesis is used to measure gas concentrations in a three-dimensional space.

With this algorithm voxels crossed by a line in a discretised environment can be identified. A disadvantage is that only one voxel per step along the driving axis is identified. This results in missing voxels that are crossed for a short distance, but are suppressed by a larger distance in another voxel in the step. With another algorithm, for example, the "Fast Voxel Traversal Algorithm for Ray Tracing" [5] these additional voxels could be identified.

3. Exploration Concept

At the beginning of this section an analysis of the problem covered in this thesis is presented. Afterwards, a theoretical approach to solving the problem is explained. Subsequently, a detailed exploration strategy is presented, which utilises the approach and defines required software and hardware capabilities.

3.1. Problem Statement and Analysis

The task is to explore and map gas concentrations in a pre-defined area. The area itself is known, which implies, that an obstacle map of the area can be created in advance. Therefore, it is not required to carry sensors for mapping objects in the area or to detect and avoid objects in the movement path. This simplifies the task as all possible measurement positions are known at all times and the trajectory of the vehicles can be calculated before the vehicles start moving. Combined with the characteristics of the vehicles, a precise estimate of the path of motion can be determined.

In contrast to that, the spatial extent and concentration distribution of the plume in the area is unknown. This information shall be gathered as fast as possible, at the most informative location, taking the limitations of the vehicles and sensors into account. The UAV has a limited flight time which is further influenced by the attached payload. Consequently, a reduced payload weight results in an extended flight time. The UGV has a large battery capacity and can therefore endure longer than the UAV and carry a heavier payload. To receive back radiation from the emitted laser beam with the sensor for the gas measurement, a reflection surface is required. It must be ensured that this is the case for every measurement. As introduced in Section 2.1.2, the TDLAS-method required measurements from multiple independent angles. At the time of the mapping, the gas distribution in the inspected volume should preferably be static. Otherwise, it leads to errors in the reconstruction. Therefore, for the study in the present work, the gaseous environment is assumed to be static and unaffected over time. This is a simplification of the problem. The TDLAS method is not suited for mapping large and fast-changing gas distributions. For such a case, a different method would have to be applied and therefore is not discussed in this thesis. In typical applications like in refineries or at landfills, it is striven to achieve continuous monitoring and to map slow and constant leakage of gas over a long period of time. This case is simplified as a static plume in the area.

The task of the vehicles is the examination of the exploration area $\Omega \subset \mathbb{R}^3$. Ω can contain obstacles of different heights and shapes. The UGV is not able to go over any obstacles. The UAV is able to fly over the obstacles O with a maximum height of $H1$ but has to go around obstacles with the height $H2$. Obstacles with a height of $H2$ are taller

than the area safe to fly in or reach above the maximal permitted flight altitude. The movement of the UAV are limited to the space $A \subset \mathbb{R}^3; A \subset \Omega$. The location of the UAV can be described by the positions $\vec{a} \in A$. The UGV can move in its movement space $G \subset \mathbb{R}^2; G \subset \Omega$. The positions of the UGV is described by $\vec{g} \in G$. The allocation of the spaces is depicted in Fig. 3.1.

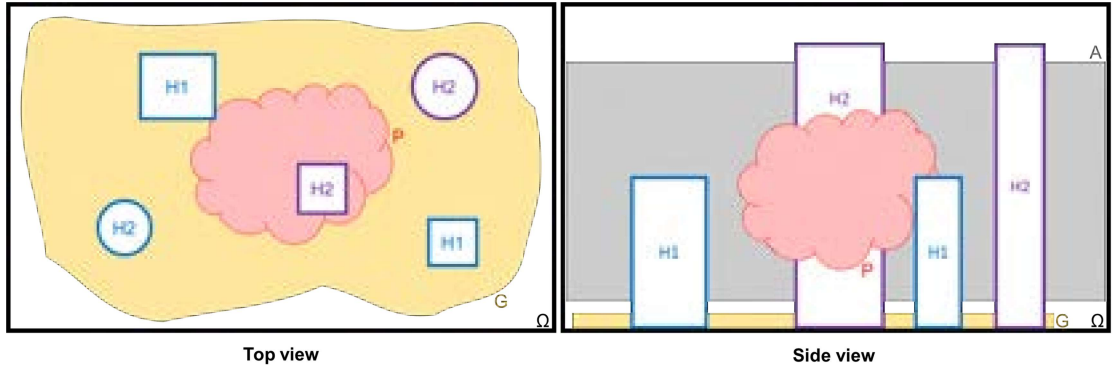


Figure 3.1.: Structure of the considered spaces. The exploration space Ω includes all other defined spaces. The movement of the UGV is limited to G while the movement space of the UAV is A . Additionally, the exploration space Ω contains different shapes of obstacles. The space P is the gas filled plume.

The area of interest is a gas plume $P \subset \mathbb{R}^3; P \subset \Omega$. It is assumed that the plume P is static and does not, or very slowly move over time.

After the analysis of the problem and the thereof resulting the broader comprehension of the problem, a solution to the problem is developed, which is presented in the following section.

3.2. Gas Distribution Mapping by Remote Sensing

For considerations and calculations performed in this thesis, the exploration space Ω , the plume P , and the obstacles O are discretised into voxels, which are three-dimensional pixels, with the edge length e_v . All voxels that are part of Ω and are not at least partially filled by obstacles, are stored in the environment vector \vec{x} . The elements of \vec{x} store the information if a voxel is filled with a concentration of the sought gas ($0 < x_{j,k,l} \leq 1$) or not ($x_{j,k,l} = 0$). A measurement at a point in time t produces the measuring result y_t . It depends on the position \vec{a}_i of the UAV and the position \vec{g}_i of the UGV. To identify the voxels that have been measured by the TDLAS-sensor, a single line matrix m_t containing a representation of all elements of \vec{x} is created. If the Direct Line of Sight (DLOS) between the UAV and the UGV crosses through a voxel, the representation in m_t is assigned the value l_{cross} , otherwise the value 0. The variable l_{cross} represents the length of the beam inside a voxel and is calculated by dividing the beam length by the number of crossed voxels. The length inside the voxels crossed by a single beam is identical due to the application

of the Bresenham line algorithm. The result of a single measurement y_t can be described by the combination of the corresponding single line matrix \vec{m}_t^T and the description of the gas distribution in the environment \vec{x} . This concludes in:

$$\vec{m}_t^T \cdot \vec{x} = y_t \quad (3.1)$$

For clarification, a two-dimensional example for a measurement is provided in Fig. 3.2.

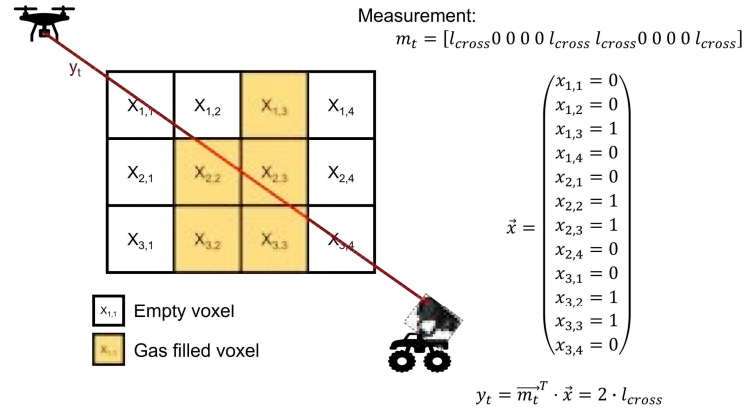


Figure 3.2.: Two-dimensional example for a TDLAS measurement performed between a UGV and UAV. Gas-filled voxels are displayed by the yellow color of the voxels. On the right side the structure of m_t , \vec{x} and the resulting measurement y_t are presented (Sensor image source [38]).

The beam created on top of the rover is emitted from the lower right corner of the area. It crosses through the voxels $x_{3,4}$, $x_{3,3}$, $x_{2,3}$, $x_{2,2}$, $x_{1,2}$ and $x_{1,1}$. But in m_t only the entries corresponding to the voxels $x_{3,4}$, $x_{2,3}$, $x_{2,2}$ and $x_{1,1}$ are filled with l_{cross} whilst the other entries are 0. As it can be seen in Fig. 3.2, it is a simplification to use the same crossing length for all crossed voxels, since the distance inside the voxels differs. This is acceptable for two reasons. First, if the voxel size is chosen relatively small to receive a high resolution of the gas concentrations, the distance in different voxels can be assumed to be similar. As a consequence of this, neighbouring voxels are expected to be filled with similar gas concentrations since there are no erratic concentration increases inside a plume. The second reason is the envisaged use of the Bresenham line algorithm as introduced in Section 2.2.2. In this case, only one voxel per step in the dominant direction is taken into account. Since the beam is a straight line with a constant gradient, the distance per step in the dominant direction is constant. In this case, only the voxels $x_{3,4}$, $x_{2,3}$, $x_{2,2}$ and $x_{1,1}$ are taken into account.

If the DLOS between the vehicles is interrupted by an obstacle, the measurement is invalid, no measuring result y_t can be obtained and m_t can not be created. Based on this knowledge, multiple measurements can be combined to solve the system of equations. The results of the measurement are combined in \vec{y} , the representation of the voxels that have been intersected by the sensors line of sight, is combined in the measuring matrix

M . In case the DLOS is interrupted by an obstacle, no new line is added to the measuring matrix M as no measurement could be performed. The principle is displayed in Fig. 3.3. In general, the correlation can then be described as follows:

$$M \cdot \vec{x} = \vec{y} \quad (3.2)$$

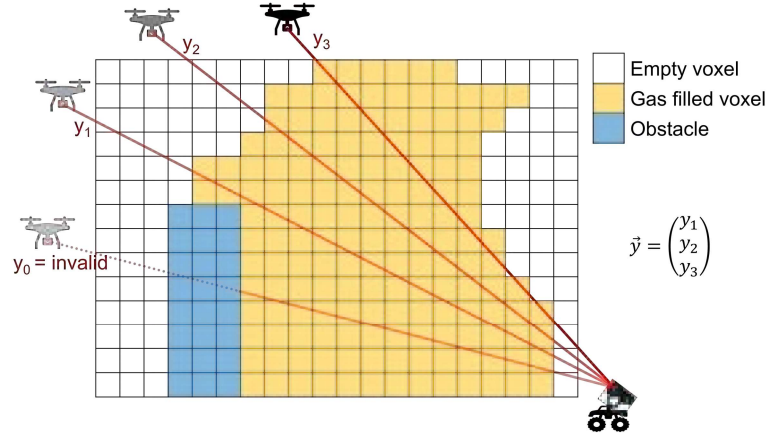


Figure 3.3.: Visualisation of the process of multiple measurements and the assembly of \vec{y} (Sensor image source [38]).

The goal of the measurements is to resolve the overall equation system and receiving the measured gas distribution \hat{x} . This is achieved at a late stage of the experiment. Therefore, as many independent measurements as there are voxels in \vec{x} have to be performed. To be able to solve the equation system from the beginning, a regularisation matrix R is stacked to the measuring matrix M . R is a diagonal matrix with constant entries on the main diagonal whilst all other entries are zero. The elements on the main diagonal are filled with the constant value r , which is called the regularisation factor. R is a quadratic matrix with the length of m_i . The measurement vector y is also expanded by a zero vector with the length of m_i to receive a solvable equation system. Thereby, an overdetermined equation system is received and can be solved like a linear least square optimisation problem.

$$\hat{x} = (M^T M + R)^{-1} \cdot M^T \cdot \vec{y} \quad (3.3)$$

With more measurements, the influence of the regularisation on the solution is reduced. A strategy is developed, to identify the measuring points which lead to the highest information gain. A measuring point is defined by a starting point and an end point of the beam. The start and the end are equal to the waypoints of the UGV and UAV during the measurement.

By utilising the waypoints of the measurements, trajectories for the vehicles can be developed. The limited dynamic of the UGV and the obstacles in the area have to be taken into account for the movement between the waypoints. For the UGV the trajectory $traj_g = \{\vec{g}_i | i = 0..t_{max}\}$ and the UAV the trajectory $traj_a = \{\vec{a}_i | i = 0..t_{max}\}$ can be created.

3.3. Exploration Strategy

Based on the results of the problem analysis and the theoretical approach, the exploration strategy is developed and presented in this section.

3.3.1. System Concept

As shown in Section 2.1, remote sensing has benefits compared to an in-situ measurement. For reasons pointed out in that section, the remote sensing method is a convenient approach for the studied task and therefore chosen for exploring the gas concentration in the area of interest. To measure the actual concentration of the gas distributed in the area, the concept of TDLAS introduced in Section 2.1.2 is used. To improve the measuring capabilities, the measurement is performed by two partners - one being the airborne UAV and the other one being the ground-based UGV. One carries the TDLAS-sensor. The other one is equipped with a retro-reflective surface. The material has the characteristic of always returning the light in the direction it originated from. Thereby, the independence from the need for the presence of a reflective surface in the area is created. In addition, the reflectivity of the surface reflecting the laser beam is known and reproducible values can be taken into account for the evaluation of the measured concentration value. Otherwise, further compensation for the reflective characteristics of a surface in the environment has to be included.

To achieve a highly autonomous process, unmanned vehicles shall be used for the exploration of the area. This takes advantage of the benefits of unmanned exploration as presented in detail in Section 2.1.3. An approach based only on ground-based vehicles is limited to measuring gas concentrations in low altitudes or otherwise, is depending on available reflection surfaces in the area. A reflective surface mounted to another ground-based vehicle allows only a sort of extended two-dimensional measurement in a plane close to the ground. If the sensor and the retro-reflective surface are mounted to two UAVs, the measurement of gas concentrations above ground level up to the maximum measuring distance of the TDLAS-sensor is enabled. Furthermore, it adds a whole new measurement direction and enables the measurement from multiple different angles, not limited to a single plane. Using airborne vehicles also brings up multiple disadvantages. The flight time of a UAV is limited by its battery capacity and can only be extended to a certain amount. The heavier the payload, for example, the TDLAS-sensor, the further the flight time of the UAV is reduced. The flight time of a UAV with a higher take-off weight is influenced less by the weight of the sensor than a smaller and lighter UAV. Then again, a bigger UAV creates more turbulence and thereby has a stronger influence on the gas distribution or must be operated at a bigger distance. This problem is increased for measurements near the ground. Due to the ground effect, the movement of the air introduced by the rotors of the vehicles close to the ground is reflected by the ground and increases the turbulence further. The area will not be a free space but will be permeated

with obstacles. To perform the measurement, a DLOS between the vehicles is required. It may be necessary to enter into the volume to be measured and therefore as little influence as possible is desirable. Furthermore, a DLOS between the sensor and the retro-reflective surface is required and must also not be blocked by parts of the vehicles themselves. To cope with these problems, the combination of a ground-based vehicle carrying the sensor and an airborne vehicle equipped with a retro-reflective surface represents a good trade-off. It enables the possibility to measure the whole three-dimensional space. The sensor is mounted to the top of the ground-based vehicle. Its size and battery capacity are sufficient to carry the sensor without a significant reduction in operation time. The UAV is equipped with a retro-reflecting surface mounted to the bottom of it. This enables a DLOS from as many angles as possible. Additionally, due to the small weight of the retro-reflector, a smaller UAV can be chosen, and thereby an entry into the volume can be accepted more easily due to the lower impact by created turbulences. However, the use of a ground-based vehicle leads to a lower movement speed than a completely airborne system would have. In addition, driving through rough terrain is more difficult. A benefit is an increase in the overall operating time.

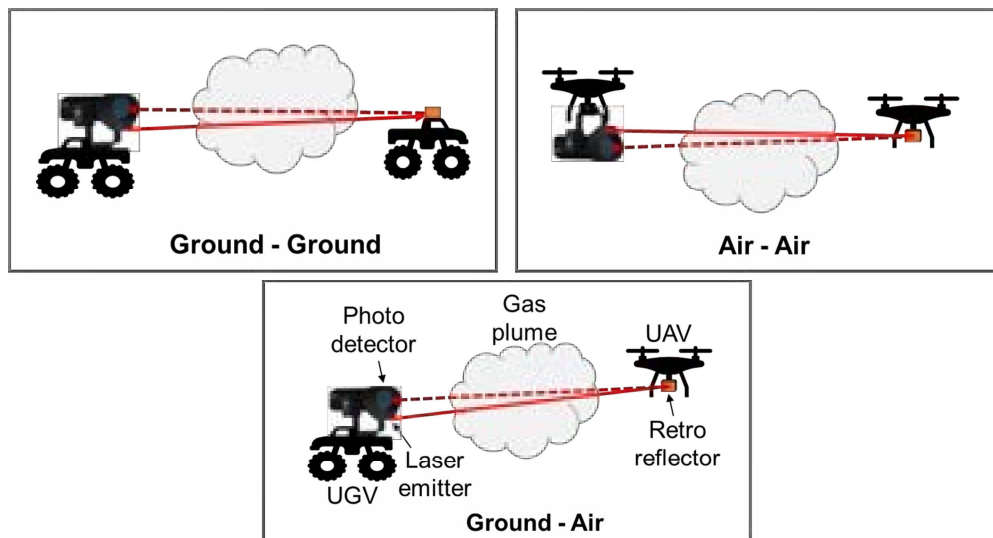


Figure 3.4.: The figure shows the different combinations of vehicle types to measure a plume with a TDLAS sensor. In the upper left corner, the measurement is performed with two ground vehicles. In the upper right, the sensor and the reflector are mounted to UAVs. In the bottom line, a cooperation of a UAV and a UGV is shown. (Sensor image source [38]).

Another benefit of the ground-based sensor is the ability to equip the vehicle with a gimbal to use it for the pointing of the sensor, without a significant impact on the maximum operating time by the increase of the weight. To point the sensor towards the retro-reflective surface, two operating steps are performed. In the first step, the UAV orients its front in the direction of the rover. Thereby, it is secured, that the landing gear of the UAV does not block the line of sight between the sensor and the reflector. At the same time,

the sensor, which is mounted on the gimbal, is oriented towards the UAV. To achieve an accurate pointing, the processing unit of the UGV needs a as precise as possible estimate of the UAV's position, to command the gimbal towards the right direction.

The goal of this combination is the operation of an autonomous system. Operators shall only be required for maintenance, monitoring, and intervening on special occasions. For this reason, a component is required to evaluate the current knowledge about the gas distribution and identify the best-suited next waypoints to perform measurements.

3.3.2. Measuring Point Selection

To eliminate the need for a central station, a processing unit is mounted directly to the UGV. A central control station has multiple disadvantages. If it is not located in the centre of the area to be investigated, which is not possible in a hazardous or dangerous environment, it needs a high-range communication system to reach the vehicles at the opposite end of the area. In addition, a clear line of sight is not guaranteed due to obstacles in the area. Depending on its capabilities, the onboard computer of the UGV can be used to run the tasks of a central station. If more computational power is required to improve the quality and the speed to achieve real-time computation, a dedicated computing unit can be added to the payload of the UGV. Adding the computation unit to the UGV's payload leads to a rather small increase in the payload weight and the energy consumption. At the same time, the communication connection to the UGV is always guaranteed because it is expected that the maximum communication range exceeds the maximal measuring distance of the TDLAS sensor. The calculation of the upcoming waypoints for the vehicles is always performed after a measurement. At this point in time, a DLOS between the UGV and UAV is assured as well. It is inevitable to carry out the TDLAS measurement. This reduces the requirements for communication equipment.

The fastest way to gain information about the area would be to perform the measurement at the location with the highest information gain. But, to identify this location, the distribution of the gas cloud would have to be known. In a simulation, this may be possible, but in a real implementation, the distribution of the plume is unknown, since it is the task to map the gas distribution. Furthermore, depending on the size of the area and the resolution of the discretisation, an enormous processing power would be necessary to identify the optimal spot in a short amount of time to enable real-time capabilities. Instead, an engineering approach is developed to identify near-optimal measuring position. It takes into account the factors that are believed to be most important to identify informative measuring targets. The approach can either be applied to a single measurement or to a series of measurements combined in a measuring pattern. These patterns are discussed closer in Section 3.3.3. The first factor is the current knowledge about the gas concentration from the previous reconstructions. A high concentration is of greater interest than a lower one. This introduces an exploitation element to the evaluation. The second part takes into account, how often a single voxel has already been measured. If it

has already been crossed by multiple beams, enough information on this voxel is available and different voxels are of higher interest to be examined. This adds an exploration part to the evaluation. As a third element, the distance to be travelled to measure at a certain point is estimated. For time and energy reasons, it is preferred not to travel huge distances without performing measurements. With these factors, a cost-function is developed. To balance the three elements of the cost function, a weighting parameter is introduced for each factor. Whilst for in-situ measurements, the parameters only have to be calculated voxel-wise, the cost function for the TDLAS measurement takes into account all voxels crossed by the beam. This results in the following cost-function:

$$K = \frac{\sum_{i=0}^n \frac{c}{C_i+1}}{n} + d \cdot D + \frac{\sum_{i=0}^n m \cdot M_i}{n} \quad (3.4)$$

The lower case letters c , d , and m are the weighting factors to proportionate the different elements of the cost function. The first addend is representing the concentration consideration. C_i is the gas concentration of the voxel i . The concentration of the voxels is normalised by $+1$ to prevent division by 0. A higher concentration leads to a lower cost. Therefore, the concentration in all voxels measured with the considered beam or pattern is summed up and divided by the number of voxels measured by it. This is necessary to compare patterns and beams with a different number of voxels.

The second addend D is the distance from the current position of the UGV to the starting point of the beam. A shorter movement distance and thereby less time before the next measurement is preferable. At the current state, only the moving distance of the UGV is taken into account. This simplification is acceptable because it is believed that the UAV moves significantly faster and has a higher turn rate. For a more precise evaluation, the required movement distance of the UAV may be taken into account as well. If the cost function is used for the evaluation of patterns, the distance D is the distance from the current position of the UGV to the centre of the pattern. This is a simplification deliberately chosen. For the comparison of the same pattern in different locations, it is only relevant how far the vehicle has to move before the pattern is started. Afterwards, the distances moved are identical. If different patterns are compared, an inaccuracy is created for which it is not accounted for. Nevertheless, the focus of the cost function is balancing between the exploiting concentration term and the exploring measuring term. The distance factor d is expected to be rather small, which minimises the inaccuracy by the simplification.

The last element of the sum is the measurement term. M_i is the number of how many times the voxel i has already been measured by a beam. The measurements of all voxels crossed by the beam or the pattern are multiplied with the weighting factor m and normalised with the number of all voxels in the considered measurements. This is necessary, for the comparison of beams and patterns with a different amounts of voxels.

Further, it is investigated in the wake of this thesis whether the performance of the cost function presented in (3.4) is improved if the variance of the knowledge about a voxel is

taken into account. To do so, a variance term is added to the function which results in:

$$K = \frac{\sum_{i=0}^n \frac{c}{c_i+1}}{n} + d \cdot D + \frac{\sum_{i=0}^n m \cdot M_i}{n} + \frac{\sum_{i=0}^n \frac{v}{v_i+1}}{n} \quad (3.5)$$

In this case, a fourth term is added to Eq. (3.4). The variance term takes into account the variance of every voxel measured. Therefore, the covariance matrix of measuring matrix M stacked with regularisation matrix R is calculated. Each element on the main diagonal of the covariance matrix represents a voxel in the area. The value of the main diagonal element i is the variance of the voxel V_i . The variance of all measured voxels is summed and normalised by the number of voxels.

With the two versions of the cost function, beams can be identified that promise to lead to a high information gain. Since the TDLAS method requires measurements from independent angles, it is reasonable to identify measuring patterns at a location and not only a single beam. In the scope of this thesis, five different patterns are being evaluated in the simulations and are introduced in the following subsection.

3.3.3. Patterns

For all patterns, the starting points of the beam can only be located in the bottom layer of the environment. This is inevitable since the UGV can only move in the base layer. This bottom or base layer is the voxels directly above the ground at an altitude of 0 m. For the UAV it is desirable to stay in the top layer of the volume, but not mandatory. If the UAV stays in the top layer, it is further away from the plume and the influence of the UAV on the plume is reduced. The top layer is the voxels on the upper end of the considered area. In Fig. 3.5 an overview on the investigated patterns is provided.

- Hourglass: For this pattern, the UAV moves only in the top layer. The pattern consists of nine beams. Eight of these beams have starting and ending points that are located on the edges of a rectangle. One in each corner and one on each median of the edges. The ninth point is located in the centre of the rectangle. By always positioning the vehicles on the opposite side of the rectangle in their layers, the measurements result in an hourglass-like shape.
- Cross: The cross is a simplified version of the hourglass in which only measurements parallel to the x- and y-axis of the coordinate system are being performed. The four diagonal measurements of the hourglass pattern are omitted which results in a total of five measurements between each reconstruction of the plume.
- Inverted Pyramid: In this pattern four beams are used. The UGV moves to a point in the centre of a rectangle. The UAV flies to the corners of the rectangle and measurements are performed there. The UAV only moves in the top layer of the environment.
- Pyramid: The pyramid is similar to the inverted pyramid. In this pattern, the UAV stays in the centre of the rectangle in the top layer. The UGV moves to the corners

of the rectangle in the bottom layer. Resulting, four measurements are performed during the pattern.

- **Cube:** In this pattern, the UGV stays still at a position in the centre of a rectangle in the base layer. This time the UAV is allowed to enter into the volume. The rectangle is expanded to a three-dimensional cube. The UAV flies into every voxel on the four sides and the ceiling of this cube. Thereby, during this pattern measurements are performed from many different angles. This pattern is used to investigate the effect for a multitude of separate measuring angles. It is to be determined if the information gain can be increased.

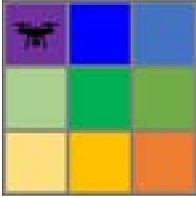
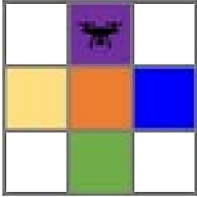
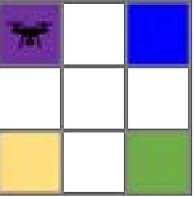
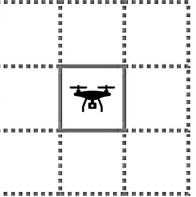
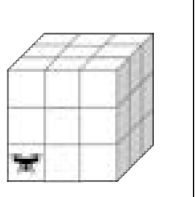
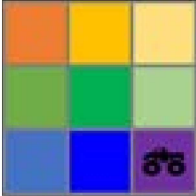
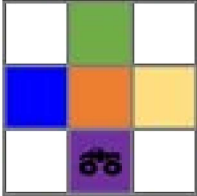
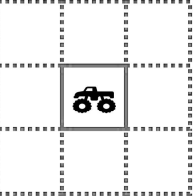
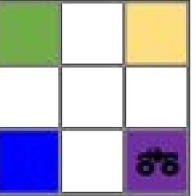
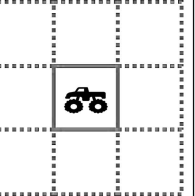
	Hourglass	Cross	Inverted Pyramid	Pyramid	Cube
UAV					
UGV					

Figure 3.5.: The figure shows an overview on the measurement positions for the different patterns.

In the upper row the position of the UAV is shown and in the lower row the position of the UGV is depicted. Identical colors in the top and bottom rows of the same column indicate the position of the UAV and UGV at the same point in time.

In addition to the five patterns, it is also investigated to reconstruct after each measurement and then choose the best possible beam calculated by the cost function. As for the previous patterns, limitations apply to the creation of a beam. The starting point of the beam can only be located in the lowest layer of the volume since the UGV is only able to move on the ground. As it is preferable that the UAV does not enter the plume, the end of the beam is only allowed to be in the top layer of the volume. Additionally, this restriction reduces the calculation time significantly, by reducing the number of possible beams. Nevertheless, the calculation time is still drastically more than for the pattern approach. Still more variants have to be considered, resulting in a time-consuming process. The presented approach uses only a single beam. To reduce the computational expense, it could be modified to determine the best n-beams. Thus, more measurements are conducted and the reconstruction and target calculation is performed less often. In the scope of this thesis, only the variant with the single beam is examined.

3.3.4. Communication

For the communication between the vehicles and the ground control station of the operator, a line of communication is required. The operator requires knowledge about the current health status, position and targets of the vehicles to intervene in case of malfunctions or emergencies. Furthermore, a communication connection is required to send the target locations and to verify the position of the UAV before a measurement is performed. Depending on the local conditions, multiple solutions can be taken into account. The benefit of the need for a DLOS for the measurement also provides a DLOS for data communication between the vehicles. During the movement, this line might be interrupted, but it is guaranteed it is recovered latest at the target position.

In regions with good coverage of mobile networks, it would be possible to communicate by mobile data. Although, it is simple to provide the vehicles with a sim-card based connection to the network of a mobile provider. Most of the time these devices are not receiving a dedicated IP address so it would not be possible to directly address the vehicles, unless they are connected to a server or to a virtual private network. Therefore, this requires additional infrastructure and still depends on the availability of mobile internet which can not be influenced directly.

Another possibility would be the use of a Wireless Local Area Network (WLAN). This is only possible when the area that shall be explored is not too large. Due to the limited range of the sensor, it can be expected that the maximum distance between the UAV and the UGV does not exceed 150 *m*. This distance can still be covered with decent WLAN antennas.

Since the messages are mainly telemetry information and few commands for waypoints, a radio solution would also be possible. The amount of messages and size is similar to messages used to control unmanned vehicles with a radio-based remote control. The only disadvantage is the limited range to the operator station. But for the communication between UGV and UAV under the given constraints listed above, it would be sufficient.

3.3.5. Extension Capabilities

The prior introduced concept can be scaled up without high efforts. It can be done in different manners. One option is the combination of multiple pairs of UAV and UGV. The area has to be segmented into multiple sections. Each vehicle pair is assigned to a section to map the gas concentration inside. In a less segmented approach, the area is not divided into sections. Instead, all vehicle pairs survey the whole area together. When new points of interest are identified, they are assigned to the closest pair of vehicles. In the last step, the concept can be adapted to a more swarm-like behaviour. The fixed bond of the vehicle pairs is dissolved. Each UGV can interact with every UAV. This increases the flexibility and most likely the efficiency of the swarm. At the time a new line to be measured is identified, the closest and earliest available UAV and UGV are identified and assigned to perform the measurement. In case the exploration concept is extended to more vehi-

cles, a communication solution with a wider range and a capability to communicate with multiple vehicles at the same time is required. The radio-based solution would no longer be sufficient for this option.

4. Implementation

For the implementation of the developed exploration concept, a multi-step approach is chosen. To verify the feasibility of the concept, a Python simulation is implemented to evaluate the capabilities of the concept. The simulations allow a faster than real-time runtime which enables a more convenient examination of various aspects and parameters influencing the concept. Once the concept is verified and an acceptable set of parameters is identified, a second simulation utilising software components ROS and Gazebo is implemented. This simulation is based on software-in-the-loop-simulations of the real vehicles and enables physical interactions. It is expected to increase the knowledge about the behaviour of real vehicles. In the last step, the developed software is transferred to the actual hardware (i.e. UGV and UAV) and a proof-of-concept demonstration is performed. During the demonstration the sensor and the plume are simulated.

At the beginning of this section, a short overview of the devices foreseen for the hardware, implemented and assumed for the simulations, is provided. Afterwards, the different stages of implementation are presented. At the end of the chapter, a description of the planned proof-of-concept demonstration (i.e. field test) is outlined.

4.1. System Setup

The main hardware components are the ground and the air vehicle. In both cases, already available vehicles in the institutes are used. The Holybro S500 (UAV) and the Summit XL (UGV) are used in the simulation and in the proof-of-concept demonstration. Communication solutions already in use by the institutes are utilised. Further, a gimbal and a TDLAS sensor are required to realise the concept. Their functions are simulated as well as for the sake of completeness of the concept, recommendations for these devices are provided.

4.1.1. Airborne System

The airborne system is a small and light quadcopter model used by the Institute of Flight Guidance called *Holybro S500*. It uses a *Pixhawk 4* flight controller. On the flight controller, the flight management software *PX4* is executed. The Holybro S500 is a highly customisable construction kit UAV. In the current case, a 4S lithium polymer battery with a capacity of 5500 mAh is used. This enables a flight time of 15 to 20 minutes depending on the environmental conditions. In general, it is possible to use different kinds of batteries which increase the flight time. It has to be taken into account that at a certain point the increased weight of the larger battery begins to reduce the flight time. The S500 has

a size of approximately 390 mm by 390 mm and a height of 240 mm . With its size, the S500 is an ideal fit for a small UAV and still capable of carrying a retro-reflector. For position determination the UAV is equipped with a small Global Positioning System (GPS) receiver. To communicate with the flight controller for monitoring and commanding, the *MAVLink protocol* is used. It is a communication protocol widely used for custom build UAV's. For the communication connection, a 433 MHz radio solution is used. To create the commands for the UAV and read the messages sent by the UAV, the *mavros* library, a bridge between MAVLink and ROS is available.



Figure 4.1.: An image of the Holybro S500 used for the UAV tasks taken during the proof-of-concept demonstration.

For the simulation, a software-in-the-loop-simulation of the PX4 software is provided by the developer. With a modification, it is possible to integrate it into the ROS-Gazebo environment. For the visualisation in the Gazebo environment, an exemplary small UAV model is used.

4.1.2. Ground System

The ground vehicle is a customised version of the rover Summit XL manufactured by the company Robotnik and operated by the Institute Communication and Navigation. It has a size of $720\text{ mm} \times 614\text{ mm} \times 416\text{ mm}$ and can carry a payload of 65 kg . This enables the rover to carry a TDLAS sensor and the required gimbal to point it in the direction of the UAV. The vehicle has an operating time of around 10 hours with a single battery and without carrying any payload. This is a multiple of the operating time of the UAV. Whilst the UAV can perform omni-directional movement, the UGV is restricted in its movement. The Summit XL has an all-wheel drive allowing it to move through rough terrain. The wheels can be controlled separately, allowing the Summit XL to turn on the spot. Lateral movements are not possible in the standard configuration. Special wheels, so called mecanum-wheels, are available for the Summit XL. However, these wheels can only be used in flat terrain and are therefore not applicable to many use cases outdoors. The Summit XL can be directly commanded by ROS messages. To communicate with the vehicle, a local WLAN network is used. The access point for the network is mounted at the

base station of the UGV. Devices willing to communicate with the UGV have to be connected to the network and can then monitor and command the vehicle by ROS messages. In addition to that, the base station hosts a Real Time Kinematic (RTK)-GPS system. It is used for high precision positioning of the Summit XL. To include the UGV into the ROS-Gazebo simulation, a simulation of the Summit XL is provided by the Institute of Communication and Navigation.



Figure 4.2.: The figure shows the modified Summit XL of the Institute for Communication and Navigation used for the proof-of-concept demonstration.

4.1.3. Sensor System

On the market, various sensors for different applications for TDLAS measurement are available. One use case is to measure gas quantity and concentration transported inside of pipes. The sensor range from small sensors to be integrated for industrial process control [13] to larger gas process analysers for multiple gases [14] or even large devices for integration in industrial processes or lab applications [47]. Sensors designed for the use in a pipe are not capable to measure over a great distance and therefore not suited for the present use case. A particular, smaller category of devices must be looked at for outdoor measurements. The technological progress allowed the downsizing of the TDLAS technology which enables small devices to be used as hand-held like the RLGD-100 produced by Focused Photonic Inc. [16]. The sensor is specified with a measuring range of 30 m and is calibrated solely for the measurement of methane. The maximum distance depends on the reflectivity of the object the device is pointed to, and how much of the emitted beam is returned to the sensor. It is specified with a measuring step size of 1 ppm – m with an accuracy of 10 % and a response time of 0.1 s. The measuring range is 1 – 9999 ppm – m. With modifications, this device allows to mount the sensor to a UGV. With a weight of less than 1500 g, the sensor can be transported by the UGV. By now, the first devices intended to be mounted to aerial vehicles are available. Exemplary the TDLAS of Infrared Cameras Inc. [23] and the U10 of DJI Enterprise [15] have been examined.

The TDLAS sensor distributed by Infrared Cameras Inc. is shown on the left of Fig. 4.3. It has a measuring range of $0 - 40.000 \text{ ppm} - m$ with a minimum resolution of $1 \text{ ppm} - m$ and a response time of 0.5 s [23]. The detection distance is between 0.5 m and 50 m . The data sheet states that a long-range version is available which then would not be able to detect gas concentrations in the close range between 0 m and 20 m [23]. It is solely specified to detect methane. The total weight of the sensor is 668 g . The dimensions of the sensor are $120 \text{ mm} \times 97 \text{ mm} \times 97 \text{ mm}$ [23]. With these dimensions and weight, it could either be mounted to the UGV or the UAV.

The U10 of DJI is depicted on the right side of Fig. 4.3. It is designed to be attached to a UAV to measure methane concentrations. The sensor can detect concentrations between $0 - 50.000 \text{ ppm} - m$ with a response time of 0.025 s and a sampling rate of 500 kHz . It has a weight of 700 g with dimensions of $174 \text{ mm} \times 89 \text{ mm} \times 163 \text{ mm}$. The maximum detection distance is specified at 150 m .



Figure 4.3.: On the left side the TDLAS sensor of the Infrared Cameras Inc. is shown [24]. On the right side, the sensor U10 distributed by DJI is shown [38].

The analysis of available sensors shows that detection distances of up to 150 m can be obtained. The detection distance depends on the reflecting material. The reflectivity of the material used at the UAV is not known in detail. To ensure the quality of the measurement and to apply a safety margin, the longest examined distance should not exceed 100 m . The weight of both analysed sensors is around 700 g . This weight can be carried by the UGV and does not lead to noticeable restrictions.

To point the TDLAS sensor towards the reflecting surface at the UAV, the UGV must be equipped with an additional tracking system. On the market, a broad variety of gimbals for aerial vehicles are available. There are gimbals available from low-cost solutions for amateur photographers up to high-end solutions used in the film industry. To provide an impression, in [50] an overview of the products offered by the manufacturer Gremsy is given. The Gremsy T3V3 [19] is capable of carrying up to 1700 g payload in a payload volume of $152 \text{ mm} \times 100 \text{ mm} \times 120 \text{ mm}$. The gimbal itself has a weight of 1200 g . It is a gimbal with three controllable axes. It enables a rotation range of $\pm 345^\circ$ around the pan axis, $\pm 120^\circ$ around the tilt axis, and $\pm 45^\circ$ around the roll axis. With a modified mount, it is possible to mount it to the top of the robot. The system is shown on the left side of

Fig. 4.4.

Due to the application, a two-axis camera gimbal is sufficient. Furthermore, there are no strict weight limitations due to the high payload capacity of the UGV. Therefore, systems being used for antenna tracking can also be considered. An alternative in this area can be the X-GRST-E Series by Zaber [57] which is shown on the right side of Fig. 4.4. It is a two-axis gimbal with a 300 mm payload clearance being able to carry payloads up to 15 kg. Along both axes, the mechanism can be rotated the whole 360°. The gimbal has a weight of 13.2 kg and therefore is significantly heavier than the T3V3.

The research on available gimbal systems on the market shows that no limitations are caused by the choice of the gimbal system. For this reason, no boundary conditions must be considered during the simulations.



Figure 4.4.: On the left the 3-axis gimbal T3V3 by Gremsy is shown [19]. On the right side, the X-G-RST300-E03SR10 of Zaber is shown [57].

To determine the required orientation of the sensor, precise identification of the position of the UAV is required. The reported GPS position of the UAV provides a rough estimation of its position. However, the inaccuracy of GPS is too big to point the beam of the sensor towards the UAV. Hence it is suggested to complement the system with an optical sensor for tracking the UAV. This can for example be a camera with image processing software or a sensor sensitive to a marker mounted to the UAV.

On the bottom side of the UAV, a panel made out of a retro-reflective material is mounted. These materials have the effect of returning most of the light directly back to the direction it came from. These materials are for example used in road markings or on vehicles. The panel is mounted facing the front of the UAV with a small tilt to the ground. It does not require a dedicated gimbal because it is believed that the orientation of the UAV towards the UGV is sufficient. Still, if possible, it is better when the material is applied to a semi-sphere because the influence of the orientation of the UAV is reduced.

4.1.4. ROS

ROS is a software framework to control a network of robotic vehicles, sensors, and systems supported by the *Open Source Robotics foundation* [44]. Its modular design enables users to

choose and include only required components. A ROS system is composed out of several software nodes interconnected by the ROS-network [43]. In this thesis, when the term ROS is used, it is referred to ROS1. ROS2 is a newer version of the ROS framework. Systems used for this thesis were developed with ROS1. Upgrading to the new ROS-version is possible but no simple process. Existing software has to be adapted and tested which would not have been possible in the time frame of this thesis. Therefore, it was decided to perform the development and testing with ROS1 and postpone the upgrade to a later point in time. To support the upgrade later, during the development of the new software components are preferably packages used that can be upgraded to ROS2 easily.

In the network, a master node called *roscore* is used to manage the components and the communication within the network. Initially, new nodes register at the roscore. Afterwards they communicate directly with each other. The communication is performed by topics [43]. A topic contains a message with predefined parameters and is registered at the master node. Every node in the network can function as a publisher as well as a subscriber. Publishers send messages on a topic whilst subscribers listen to topics to gain information from the received messages. Multiple nodes can publish and subscribe to the same topic. In addition to permanent processes performed by nodes, so-called services can be called to perform single-use tasks [43]. In Fig. 4.5 the working principle of the ROS-framework is shown.

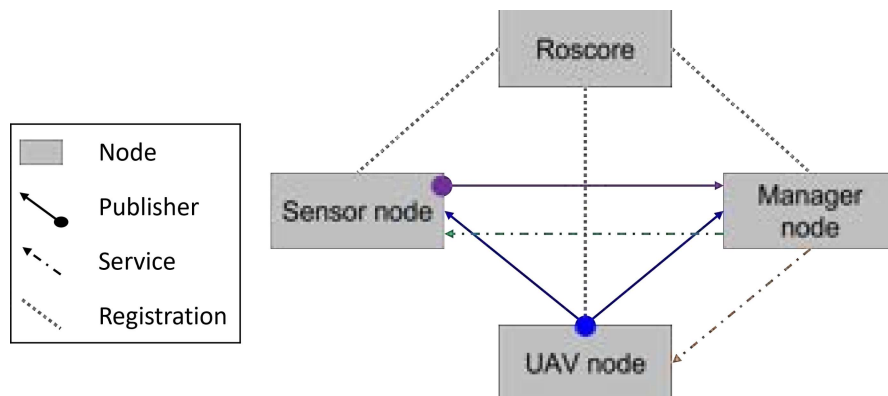


Figure 4.5.: The figure shows the principle of a small ROS framework. Three nodes are registered at roscore. Overall, two publishers and two services are used.

In the schematic in Fig. 4.5, the three nodes (i) manager, (ii) UAV, and (iii) sensor are registered at the roscore. The registration to the roscore is marked by the grey dashed lines. The manager node is the central node performing target position determination, the reconstruction, and managing the system components. The manager uses a service to command the UAV to a specific position. The UAV is publishing its position on the blue topic to which the manager and the sensor are subscribed. Once the UAV reaches its target position, the manager commands the sensor to perform a measurement. The sensor publishes the measurement by the purple publisher to which the manager is subscribed.

For monitoring and managing a ROS-system, knowledge of available topics and services

is required. Nodes registered to the master are not required to run on the same machine is a huge advantage of the ROS framework. A network with a multitude of machines can be set up as long as they are connected to the master node [44]. This enables the control of multiple vehicles and sensors. In the following, the setup used in this thesis is described. An overview of the setup is displayed in Fig. 4.6

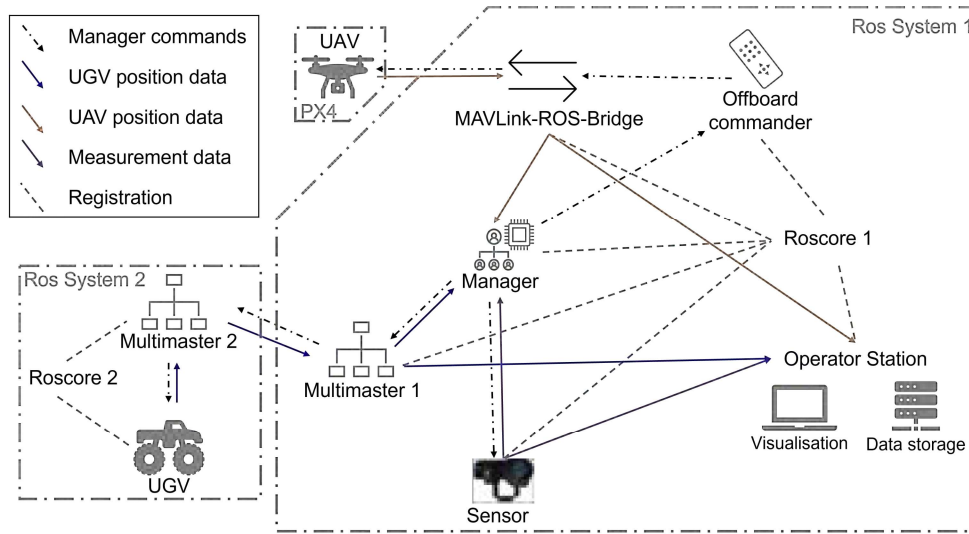


Figure 4.6.: The figure shows the structure of the ROS network used during this thesis. The network contains a UGV, a multimaster, a UAV, a MAVLink-ROS-Bridge, an Offboard commander, a sensor, a manager, and an operator station for monitoring and data storage (Sensor image source [38]).

In the figure, it can be seen that the system is divided into two ROS systems and a PX4 system. In the first ROS system seven different nodes are registered at the roscore 1. The central manager node is managing the vehicles and the sensor. Additionally, it is applying the heuristic for the target allocation and performs the reconstruction of the estimated plume. To do so, the manager uses commands to send the vehicles to their destination. Since the UGV is different hardware than the manager node, the so-called multimaster is required to connect the two ROS systems. The software is used to connect ROS environments running on different machines and forwards selected topics between the ROS systems. On the UGV a second roscore is running. To the roscore 2 the software of the UGV and a second multimaster are connected. The commands sent by the manager node and the position data sent by the UGV have to be passed on by the multimaster.

The UAV itself is not part of the ROS systems. It is a single PX4 system. The UAV is connected by the MAVLink-ROS-bridge. All commands sent to the UAV and the data received from the UAV are forwarded by that bridge. The received position data from the UAV are directly published to the ROS network by the MAVLink-ROS-bridge. The commands sent by the manager node have to be edited by the offboard commander at first. This is necessary due to the method of controlling the UAV closer described in Section 4.2.6. Once both vehicles reached their target location, the sensor is commanded

by the manager node to perform a measurement. The measurement data are published back to the manager. At all times, the position data of the vehicles and the measurement data, published on their topics, are subscribed by the operator station. There the data is visualised and stored. This design can easily be extended to multiple groups of vehicles and operators.

4.2. Simulation

The simulations are divided into two parts. At first, a more simple but faster Python simulation is performed to evaluate the concept. In the second step, a physics simulation is used to investigate the implementability of the concept and the modification needed for the transfer of the concept to a hardware realisation. Before the two simulations are described more closely, a short introduction to set assumptions and definitions for the simulations are presented.

4.2.1. Assumptions

Due to the complexity of the topic, the available hardware, and the limiting time frame, multiple assumptions are made to simplify the problem.

For the simulation and the demonstration, it is assumed, that a communication system is available that enables the communication between the vehicles and the monitoring base station. The communication system has capabilities to cover the whole area to be investigated.

It is expected that a pointing device (i.e. gimbal) is available which is capable to point the sensor in the direction of the UAV with a sufficient accuracy to hit the retro-reflector mounted to the UAV with the laser beam from the sensor. In addition, it is assumed, that a TDLAS sensor for measuring the gas concentration of the plume is available and mounted to the UGV. The sensor is capable to create beams that can measure over the maximum possible distance in the area.

Furthermore, it is assumed that the investigated plumes in the area are constant and not altered by environmental influences.

4.2.2. Coordinate Systems

For the creation, calculations, simulations, and demonstration, multiple coordinate systems have to be taken into account and a transformation between the different systems is required. In the following, the different systems and the transformation between them are introduced.

The first coordinate system is the depiction of the environment. It is a continuous three dimensional-system, characterised in metric units. The axes are x_e , y_e , and z_e . The UGV and the UAV can perform the continuous movement without being limited to any grid.

This is not only the case for the realisation with the actual hardware but also valid for the experiments performed in the simulation. The investigated area taken into account for the thesis has a size of 20 m by 20 m with a height of 10 m .

The simulation environment is created based on a two-dimensional grayscale image. The grayscale image is utilised because it can be used as the basis for the creation of a three-dimensional simulation environment as well as read into a NumPy array. Furthermore, it enables fast generation of different simulation environments without great effort. The process for the creation of the environments is described in Section 4.2.3. The image has a size of 200 by 200 pixels. The origin of the image is in the upper left corner, the x_g -axis runs horizontally to the right while the positive y_g -axis direction is going downwards. The z_g -dimension of the three-dimensional space is represented by the gray value in the image. In the creation process, the required factor between the two-dimensional image and the three-dimensional simulation environment is calculated. For the simulation environments used in this thesis, the factor is set to 0.1 to achieve an area size of 20 m by 20 m . Therefore, each pixel is representing an area of 0.1 m by 0.1 m and the gray value has to be divided by ten to obtain the resulting height in meters. The two coordinate systems are shown in Fig. 4.7.

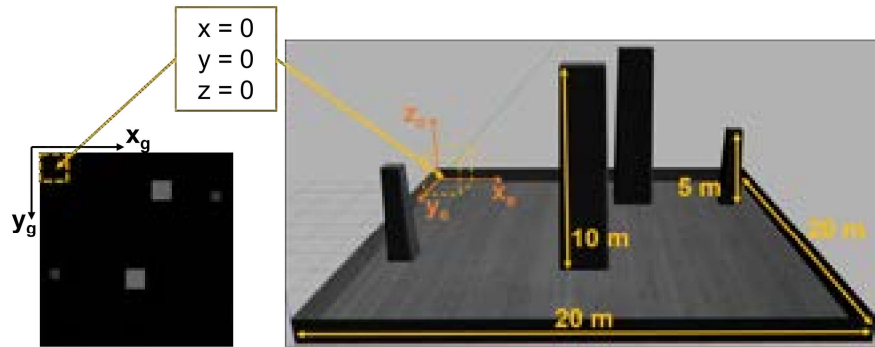


Figure 4.7.: The left side shows the grayscale image on which the simulation environment is based. The darker the color, the lower is the resulting object in the generated three-dimensional environment. The right side shows the resulting environment in Gazebo.

The environment is used as an input for the measurement algorithm. This is necessary to evaluate if a measurement is successful or if the beam gets shaded by an object. Therefore, the environment has to be transformed into a voxel-based coordinate system. A scaling factor is used based on the voxel size. Voxels are square-shaped with an edge length of 1 m . Hence, the scaling factor of 10 for transforming pixels into voxels is applied.

Each voxel is assigned with the three-dimensional coordinates x_v , y_v , and z_v as well as a number e . The coordinates are representing the position in the pixel grid. The origin is located at the lower left corner at the back of the grid. This enhances the comparability with the environment. Furthermore, the voxels receive the number e beginning at 0. The enumeration begins at the origin voxel along the x_v -coordinate, followed by the

y_v -coordinate, and ends following the z_v -direction. The definition and allocation of the voxel coordinate systems is shown in Fig. 4.8.

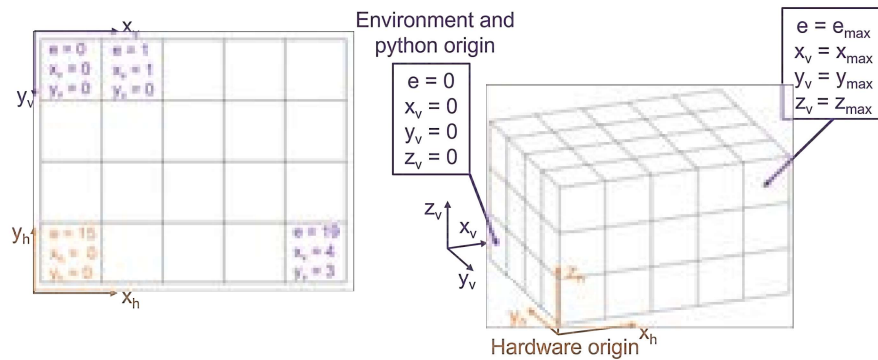


Figure 4.8.: In the figure, the difference between the environment coordinate system (purple) and the hardware coordinate system (orange) is displayed. The left side shows the two-dimensional enumeration of the voxels. On the right side, the process is continued into the three-dimensional space. Each voxel is assigned an x_v -coordinate, y_v -coordinate, z_v -coordinate, and a number e . The enumeration is identical for all coordinate systems.

Whilst the coordinate system of the vehicles in the Python simulation concurs with the coordinate system of the environment, the coordinate system of the vehicles in the physics simulation and the real hardware deviates. This is inevitable due to external constraints. During the physics simulation and the hardware demonstration, the same coordinate system is used for the vehicles. The origin of the coordinate frame is located in the lower-left corner of the map and the x_h -direction, as well as the z_h -direction, is concurrent to the prior introduced coordinate systems. Though, the direction of the y_h -coordinate is opposite to the prior introduced direction. Therefore, a transformation for the y-axis coordinates has to be performed as a function of the size of the area. The different coordinate systems are displayed in Fig. 4.8.

4.2.3. Scenarios

In the following subsection, an introduction to the scenarios used in the simulations is provided. The Python and the physics simulation with ROS and Gazebo are based on the same scenarios. For both simulation types, the surrounding environment is created based on grayscale maps. In the Python simulation, the grayscale maps are used as input for a matrix, whilst for the simulation in Gazebo, a 3D-mesh is created based on the grayscale maps to function as an environment. In both cases, the simulated plumes are read from NumPy-files in which the information about the gas distribution in the environment is stored (Appendix B).

Surroundings

To perform simulations, a surrounding environment is required. It is desirable to be able to create multiple different but comparable simulation environments for testing purposes and to verify the developed software. To generate multiple maps with reasonable effort, a solution with little expense and the possibility of automation is preferable. The selected method is based on grayscale images. These can be created in drawing applications with little effort. For this thesis, a Python script to create grayscale maps containing objects with predefined shapes has been developed (Appendix B). Obstacles in the environment are represented by simple rectangles. To produce comparability between multiple maps, the forms can be parametrised. To generate a height component, the corresponding gray scale values are stored in the map. For the Python simulation, the grayscale image is read and automatically transformed into a NumPy array. For using the grayscale map in the physics simulation, a separate tool has to be used. In [1], a software solution is presented that is able to convert grayscale images into Gazebo worlds. It is called LIRS-WCT and was developed for the creation of landscapes for the Gazebo simulator for the use in simulations of unmanned vehicles [1]. The source code is available free of charge for academic use. During the process, it is possible to scale the resulting body and define a texture that is attached to it. In this thesis the LIRS-WCT is used to create the Gazebo surroundings based on the grayscale maps.

For the simulation, five different environments are used. All of them are based on

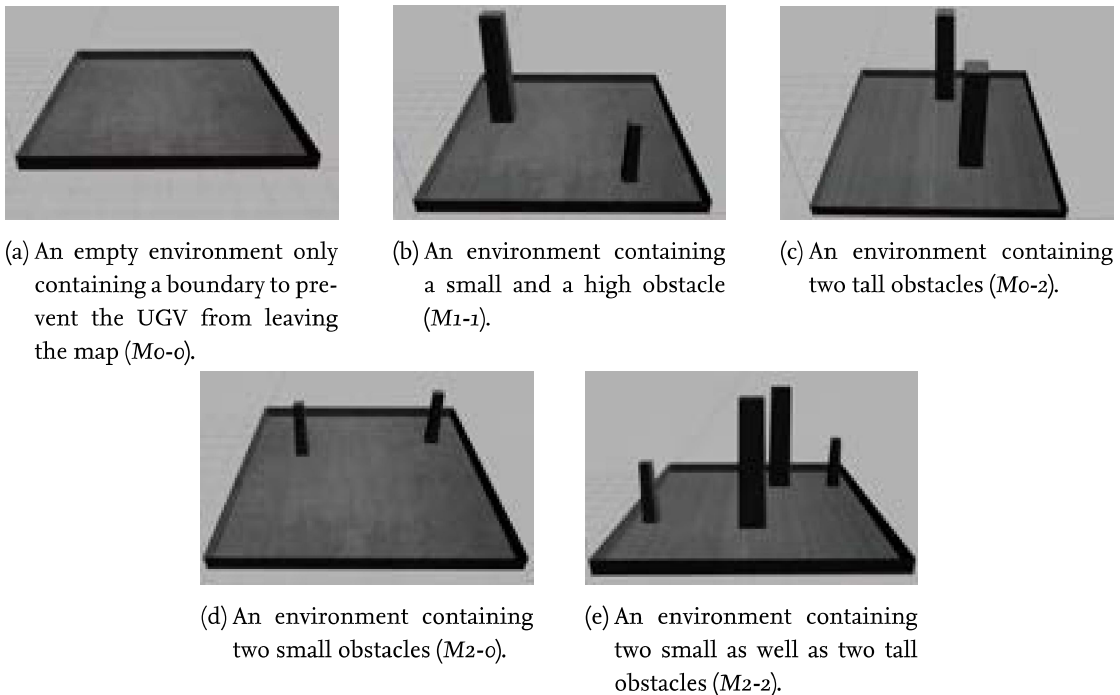


Figure 4.9.: Visualisation of the five environments used in the simulation scenarios.

grayscale maps and have a base area of 20 m by 20 m and a height of 10 m . Each environment is enclosed by a barrier to prevent the UGV from leaving the defined area. In the environments, two types of obstacles appear. The first obstacle is smaller with an edge length of 1 m and a height of 5 m . Those obstacles represent objects the UAV can cross over, but the UGV has to go around. The second type of object is obstacles with an edge length of 2 m and a height of 10 m . This represents obstacles which the UAV has to go around. To identify and distinguish the different surroundings, they are named based on the obstacles in the area. The name is composed by the scheme M "small obstacles"-"tall obstacles".

The five surroundings used in the simulations are shown in Fig. 4.9. The first environment does not contain any obstacles, just a border around the area so that the UGV cannot leave (M_0-0). The second environment contains one small and one tall obstacle (M_1-1). Environment three contains two tall obstacles (M_0-2) whilst environment four contains two small obstacles (M_2-0). The last environment contains two of each object type (M_2-2). It is believed that the mix of the environments is sufficient for a first verification of the exploration strategy.

Plumes

To increase the number of scenarios and to evaluate the heuristic for different situations, five different plumes are used during the evaluations in the simulations. The plumes are created as NumPy arrays. In a plume generator script an array with the dimensions defined in a configuration file, is created (Appendix B). Depending on the configuration, a series of entries of the array is filled with the specified value. During this thesis the concentration is considered unitless. The array is used by the Python simulation as well as by the physics simulation. The basic plume is a sphere with a diameter of 8 m and a constant concentration of 1 in the centre of the map (Fig. 4.10 (a)). At this location, the plume can be measured from all sides. The second plume is a similarly shaped plume but located in the farthest corner of the area (Fig. 4.10 (b)). Thereby it cannot be measured as often from the outer sides as the centred plume. The third plume is a combination of two spherical shaped plumes with a diameter of 8 m and a concentration of 1 (Fig. 4.10 (c)). These spheres are located on opposite corners. They represent a case in which multiple concentrations are present in an area but not interconnected. The fourth plume is again positioned in the centre of the area. Though, it is a sphere consisting out of two layers with a different concentration (Fig. 4.10 (d)). The outer layer with a thickness of 2 m has a concentration of 0.5 whilst the inner sphere with a diameter of 6 m contains gas with a concentration of 1. The last plume combines the layered plume and the partitioned plume. It consists also out of two spheres with a diameter of 8 m each (Fig. 4.10 (e)). The outer layers with a thickness of 2 m have a concentration of 0.5 and the centre a higher concentration of 1.

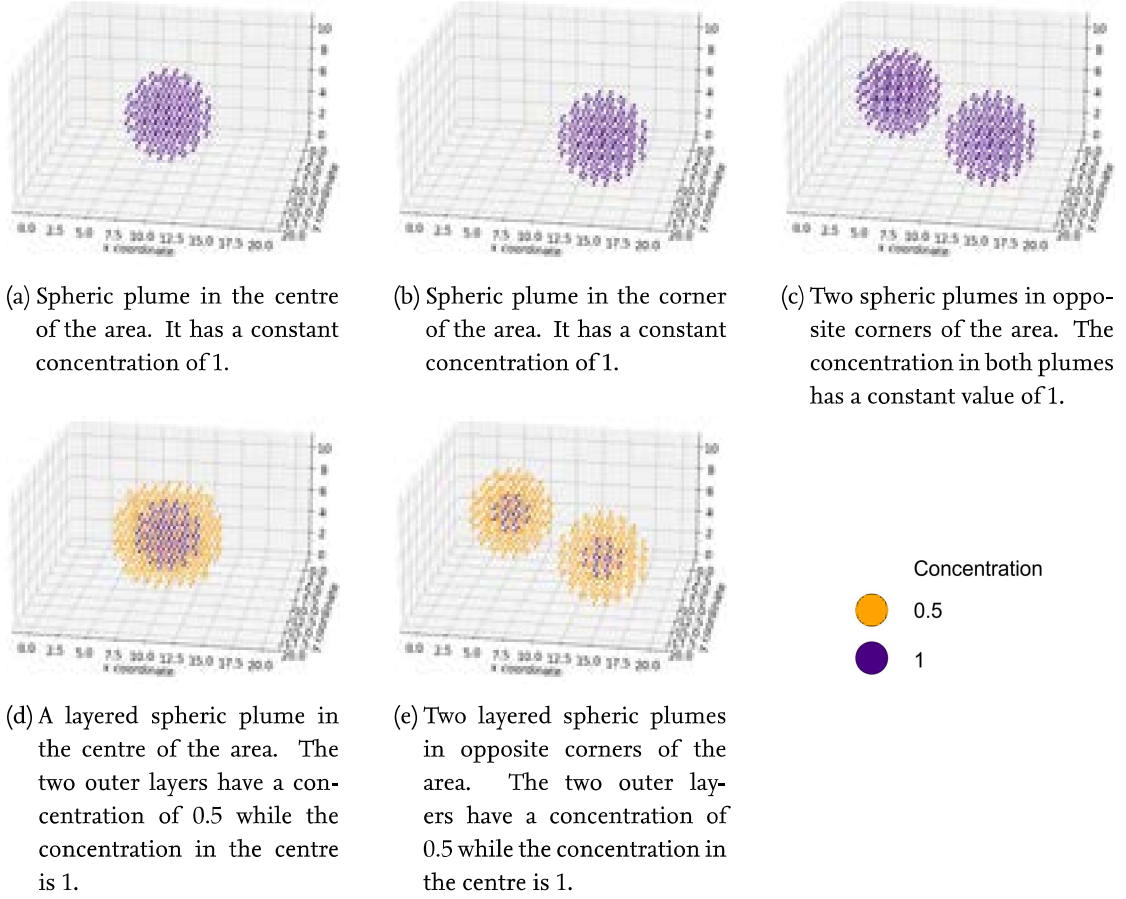


Figure 4.10.: Visualisation of the five plumes used in the simulation scenarios.

4.2.4. Evaluation Criteria

To evaluate the simulations and the success of the exploration concept, a series of evaluation criteria are defined. The main criterion is the quality of the plume reconstruction based on the measurement. To evaluate it, the Normalised Root Mean Square Error (NRMSE) is calculated as shown in the following equation.

$$NRMSE = \frac{\sqrt{\frac{\sum_{i=0}^n (\hat{x}_i - x_i)^2}{n}}}{\sqrt{\frac{\sum_{i=0}^n x_i^2}{n}}} \quad (4.1)$$

In the numerator, the Root Mean Square Error (RMSE) for the concentration in the voxels is calculated. It is based on the element-wise subtraction of the true concentration x_i from the estimated gas concentration \hat{x}_i inside a voxel. The denominator is used to normalise the result. It is the root mean square of the true gas concentration of all voxels in the investigated area. This version of the NRMSE is from here on called plume error.

In addition to the plume error, the distance travelled by the UGV and the UAV during the measurements are used as criteria. They correlate with the power consumption of the

vehicles which again influences the mission duration. Another criterion that has a similar influence on the overall situation, is the time that has passed since the beginning of the mission. Even, if the UAV is not changing its position, it consumes battery power to stay hovering in the air.

The three criteria are used to evaluate the results of the Python simulations.

4.2.5. Python Simulation

The objective of the Python simulation is to create a possibility to evaluate the design of the exploration concept faster than in real time. The simulation is used to examine the parameters of the cost functions and to identify movement and measuring patterns for knowledge gain.

The Python simulation is grouped into modules to enable modifications and to facilitate the reuse of some modules during the physics simulation. The ROS version of the hardware is not compatible with Python versions newer than Python 2.7 but ROS 2 is able to support Python 3. Therefore, during the implementation in Python, as far as possible, packages compatible with Python 2.7 as well as Python 3 are used. Since it is expected that a single simulation run takes a certain amount of time, it is striven to create a highly automatable simulation. Modifications for this purpose are not discussed in this thesis. During the simulation, logfiles containing the data required for the reproduction of the simulations are created and stored in a separate log folder for each run (Appendix B). In this folder, the configuration file, a recording of the displayed messages, a save of the created plots, and a .csv-file containing data for every beam are stored. The data file contains the beam number, the time passed since the beginning of the simulation, the distance moved by the UGV, the distance moved by the UAV, the value measured by the beam, the number of voxels crossed by the beam, the length of the beam, the error of the estimated plume at this time, the pattern used for the measurement, the position of the UGV, and the UAV.

In the following, a short overview of the steps performed by the simulations is explained. To support the understanding, the main procedure of the simulation is depicted in Fig. 4.11.

Initialisation

In the initialisation, all parameters are loaded from the configuration file. The configuration file is written in a *YAML* file. It is a format often used for configuration files since it is human-readable. Furthermore, the environment is created based on the obstacles file and the plume file provided in the configuration. Afterwards, the vehicles are configured and set to their starting position.

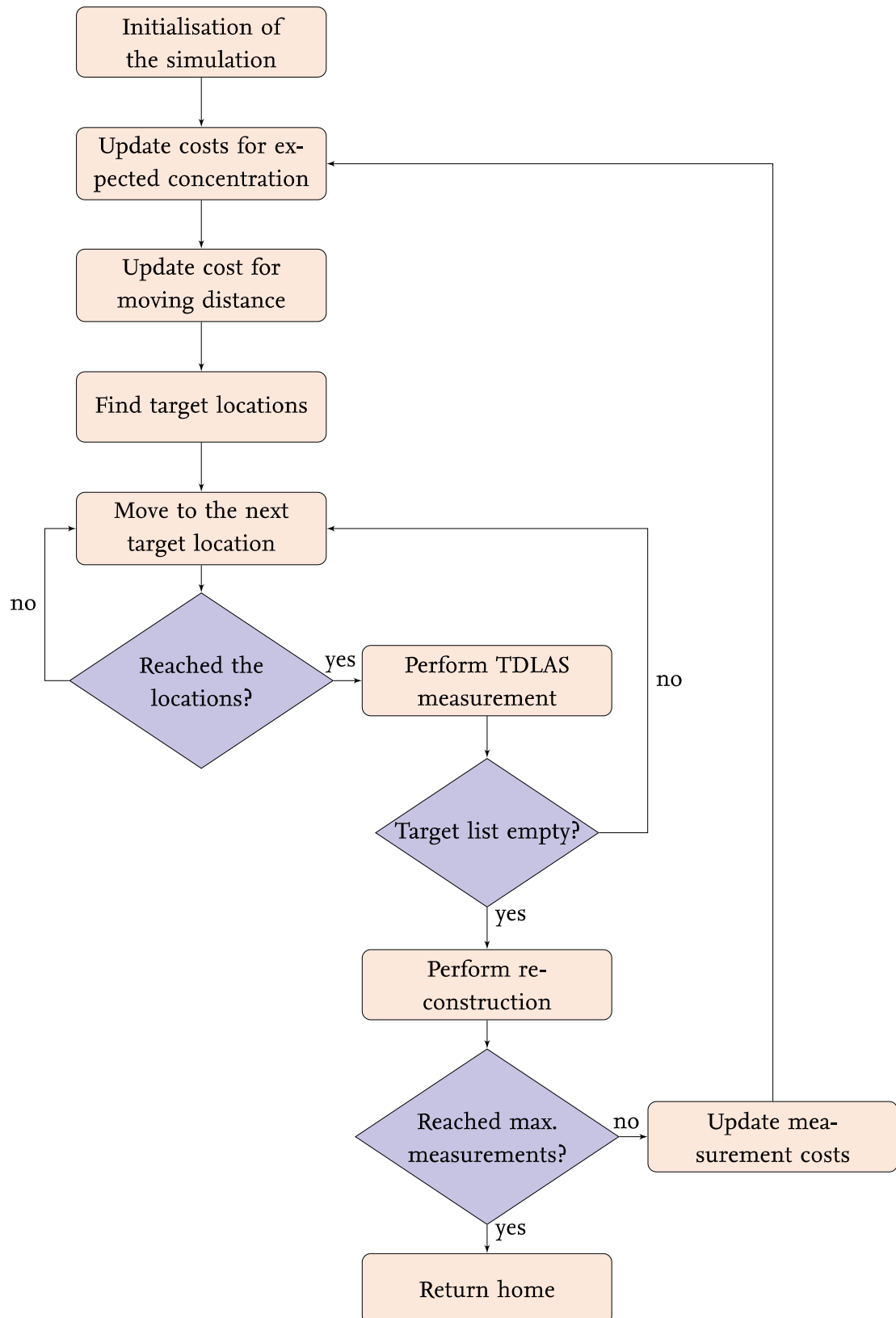


Figure 4.11.: The figure shows the procedure performed during the Python simulation.

Measurement location determination

Before determining the next suitable measurement position, the cost data is updated. The information about the current concentration in the voxels is updated after each reconstruction of the plume. It is stored in a three-dimensional array. The size of the array is based on the dimensions of the area. The measurement costs are also stored in a similar three-dimensional matrix and updated after each measurement. The third cost component is obtained depending on the current position of the vehicles. Therefore, it is identified immediately before the targets are allocated. Since the distance costs only take the UGV into account, a two-dimensional matrix is sufficient to store the information. For patterns, the distance is always calculated to the centre of the pattern. Afterwards, the distance cost is projected to all voxels of the pattern. When the target allocation process is performed, the three matrices are used to calculate the costs in accordance with the heuristic presented in Eq. (3.4). The cost calculation takes every defined pattern configuration and fits it to every possible position in the area. In the end, it chooses the pattern configuration with the lowest resulting cost and derives the position of the vehicles from the starting and end points of the beams. The list of target positions then is processed with a solver for the travelling salesperson problem. For this task, an available library from the OR-Tools is used [51]. The input for the solver is a matrix containing the distances between all points to be visited. The output is the order of the points to receive the shortest path.

Movement

With the arranged list of target points, the movement of the vehicles can be performed. The vehicles are sent to the next point in the list. To calculate the path, an implementation of the A* algorithm based on [25] is used. The movement of the UGV uses a two-dimensional version whilst the A* algorithm for the UAV is extended to the third dimension. Then the distance moving along the path is stored for each vehicle. The elapsed time during the movement is calculated based on the movement speed specified in the configuration file and stored as well. It is identified which vehicle requires a longer period of time to reach its target. This time period then is added to the estimated runtime for the simulation.

Measurement

Whenever both vehicles have reached the target location, the TDLAS measurement is performed. In the simulation, the Bresenham line algorithm is implemented for three-dimensional space and is used to draw a line from the beam starting point to its endpoint. The software takes the voxel list resulting from the algorithm and compares it with the environment NumPy array. At first, it is checked if an obstacle is hit or if it is a valid beam. If the beam is valid, the concentration of the voxels is summed and stored as the

measurement result. Furthermore, the information which voxels are crossed by the beam, the beam length, the beam starting point, and the beam end point are stored. After the measurement, the current position is removed from the target list.

Gas Cloud Reconstruction

When the target list is empty, the reconstruction of the plume from the measurements is performed. With the information on the voxels crossed by the beams, the measuring matrix is created. For each beam, it is marked in its row of the measuring matrix which voxels were crossed by it. The information is taken from the beam data. The measurement matrix is supplemented with the measurements performed since the last reconstruction and stacked with the regularisation matrix. The same steps are performed with the result vector. To solve the linear least square problem, an existing solver from the SciPy library is used. In the beginning, it is investigated to use a non-negative least squares solver [10] since the gas concentration is always positive. However, it is not optimised for sparse matrices like the measuring matrix. Therefore its performance is rather slow. Whereas a solver optimised for sparse matrices [11] generates the solution significantly faster. On the downside, it can occur that the solution contains voxels with negative concentrations. This is accepted since it occurs rarely and is corrected with the next iteration. In future work, the constraints for non-negative concentrations can be added by modifying the solver or applying a different solver. After the reconstruction is performed, the reconstructed plume is compared to the simulated plume, and the normalised root mean square error is calculated and stored to evaluate the results in the end. Then the costs are updated and the next measurement location determination loop is performed. The process is repeated until the maximum amount of measurements defined in the configuration file is reached.

4.2.6. Physics Simulation

The objective of the physics simulation is to transfer the capabilities of the Python simulation to a more realistic simulation. By doing this, it is accepted to increase the simulation time. In return, the constraints of the vehicles are taken into account in the simulation. It is strived to create a flow that is later executable in real-time. The focus of the Python simulation is the fast determination of parameters for the heuristic and the verification of the concept. The physics simulation is used to implement and debug the interfaces to the vehicle hardware. Additionally, it is used to implement the communication protocols between the different components of the concept and to test the coordination. Once the functionality of all components are verified in the physics simulation, it is easier to transfer the software to the hardware-in-the-loop system demonstrations.

Simulation Software

For the physics simulation, a combination of ROS and Gazebo is used. Gazebo is a three-dimensional simulator for robotics [17]. It is an open-source software. Gazebo provides rendering of environments based on mesh structures. It is possible to model working vehicles and sensors in the environment. Additionally, it provides an interface for ROS to control and monitor the simulated systems. In the present thesis, the UGV and UAV are simulated in the Gazebo environment. It provides physical feedback to the vehicles. The controller of the UGV is simulated in a ROS framework that is connected to Gazebo. The flight controller of the UAV is used in a software-in-the-loop-simulation which is connected to Gazebo as well as to the ROS framework. To control the vehicles, multiple nodes are added to the ROS framework. In Fig. 4.12 the Gazebo simulation with the two vehicles and four obstacles is depicted.

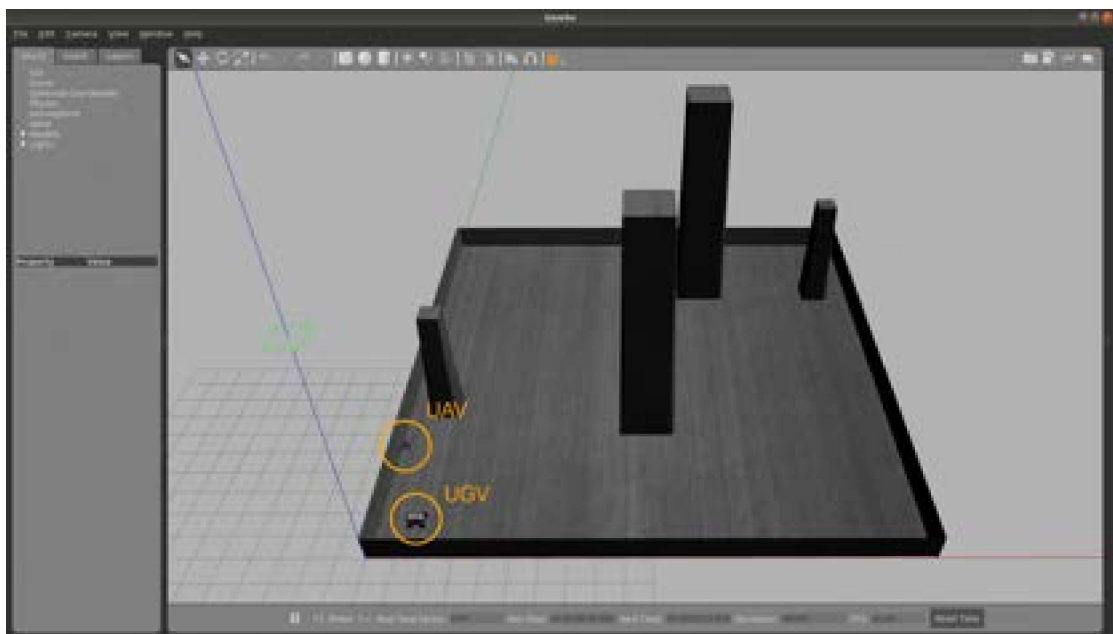


Figure 4.12.: The figure shows the graphical user interface of the Gazebo simulation. In the simulation the UGV and the UAV are located in the environment with four obstacles.

UGV Simulation

The software of the UGV consists mainly of multiple ROS nodes. Therefore, it can be executed on every Ubuntu system with the corresponding ROS version installed. To simulate the Summit XL, Robotnik provides a basic ROS and Gazebo simulation. The DLR Institute of Communication and Navigation provides an adapted version of this software. In the modified version, software nodes are added to integrate the RTK system for the positioning of the UGV. The simulated UGV is addressed with the same ROS messages as the hardware UGV. For the use in this simulation, three messages are required.

- *rtk_odometry*: Provides information about the current pose of the UGV
- *move_base_simple/goal*: Used to send the UGV to its target location.
- *move_base/result*: Used to publish a status update when the UGV has reached its commanded pose

Since the software of the Summit XL has a more advanced path planner included than the simple A* star algorithm, the A* algorithm is not used for the UGV. It is directly commanded to the measurement point. To inform the path planner of the Summit XL about the obstacles in the area, the grayscale image is transformed into a bitmap and linked in the UGV's configuration file.

UAV Simulation

The flight controller software PX4 can be executed on an Ubuntu system in a software-in-the-loop mode [7]. Furthermore, PX4 provides a implementation with an intersection to a Gazebo simulation [8]. The simulation includes a basic UAV model in the Gazebo environment. By combining this simulation with the mavros library, it is possible to command and monitor the PX4 flight controller from a ROS node. These messages are used for the simulation as well as for the communication with the actual hardware. To control the UAV via the MAVLink messages, it is necessary to set the flight controller into the so-called *OFFBOARD* mode. The mode requires a constant heartbeat with a frequency higher than 20 Hz. Once the mode is set, it is possible to command the UAV to take off. For this purpose it is necessary to set flags in the flight controller configuration regarding the offboard mode. To be used in the simulation, the controller must be configured to allow flying without a remote control and without a communication connection to a ground control station. After the takeoff, the UAV heads for target positions received via MAVLink messages. For the UAV two messages are utilised. To command the UAV to its target position, the *goal* message is required. The *local_position/pose* message is used to receive the current pose of the UAV. In contrast to the UGV, the UAV does not provide a status update upon reaching its target location. Therefore, it is necessary to constantly monitor the position of the UAV and decide when the target is reached. A discrepancy of 1 m is set as a threshold for reaching the target position. With some modifications performed during this thesis, a single simulation configuration containing both vehicles has been created. In the initial state, both simulations have a separate workspace and are configured to use separate simulation environments. It is necessary to create a reference in the workspace of the UGV to link to the workspace of the UAV. Additionally, a launch file is created (Appendix B). In this launch file at first, the Gazebo environment is launched with the UGV in it and the corresponding software-in-the-loop simulation of the Summit XL. Afterwards, the launch file spawns the UAV in the Gazebo simulation and starts the software-in-the-loop simulation of the PX4 flight controller software. The flight controller software is linked to the Gazebo simulation. In advance, it is important

to provide the flight controller software with an origin to align it with the origin of the Gazebo simulation. In the next step, the MAVLink to ROS bridge is launched. It is configured to communicate with the simulated flight controller. Thereafter, a ROS node to command the UAV to the offboard mode is launched. During the initialisation, the node commands the UAV to take off and from then on supplies the MAVLink to ROS bridge constantly with a heartbeat message.

Adaptation of the Python Simulation

The remaining functionality is adapted from the Python simulation. The overall procedure for the physics simulation is similar to the one applied in the Python simulation. The control of the simulation is performed by the *manager* node. It administers the cost arrays and performs the target acquisition. For these steps, the functions of the Python simulation are repurposed. Once the targets have been identified, the manager node sends the coordinates to the vehicles. Then the manager node waits until the vehicles have reached their targets. The sensor is simulated in a separate ROS node. The manager node interacts with it by two ROS topics. On the *vehicle_position* topic the manager sends the current position of the vehicles to the sensor. When the sensor receives this message, it performs a measurement between these two points. For the measurement, the implementation of the Bresenham line algorithm from the Python simulation is used. Afterwards, it publishes the measured value on the *measurement* topic. The manager is subscribed to the topic and waits for the response of the sensor node. When the manager receives the result of the measurement, it checks if further points are stored in the target list. If this is not the case, it performs the plume reconstruction process. Therefore, it reuses the functions from the Python simulation as well. After, the loop starts all over again until the maximum amount of measurements is reached.

4.2.7. Roadmap for the Simulation Runs

In the simulations, multiple parameters are examined to develop the best version of the heuristic to achieve the highest information gain during the mapping of the gas concentrations in the area.

In the first step, values for the regularisation factor r on the diagonal elements of the regularisation matrix R are examined. To do so, random positioning of the vehicles in the area is performed to achieve a neutral result. The only boundary conditions are that the vehicles are not allowed to be inside obstacles, the required line of sight is mandatory and the UGV can only be placed in the ground layer of the environment. The value of r is expected to influence the calculation time for the reconstruction and the influence of measurement errors. After these evaluations, one value is chosen for r which then is used for all further examinations.

With the chosen regularisation factor, measurements with a random placement in the different scenarios are performed. For the random movements, the UAV can be placed

in every voxel inside the environment that is not occupied by an obstacle. The UGV is placed only in voxels of the bottom layer of the area. In each run, the environment is measured with 4000 beams. The results created in these simulations are used as a baseline for evaluating the quality of the chosen exploration approach.

In the next step, the weighting parameters of the heuristic are studied. These are d for the influence of the distance to be moved to the measuring points, m for the influence of the already performed measurements at these voxels, c for the influence of the concentration according to the current reconstruction, and v for the influence of the diagonal elements of the covariance matrix of the measuring matrix M . All of the parameters are first examined in a wider range to identify the magnitude to be chosen. Afterwards, simulations with a smaller range are performed to identify the best performance. An analysis of these simulations is performed in Section 5.1.3.

Once the best values for the weighting factors have been identified, the placement method for the vehicles is examined. Therefore, multiple simulations with different measuring patterns as introduced in Section 3.3.3 are performed. In addition to the different types of patterns, the patterns themselves are varied in size to enable different measuring angles. The edge length of the patterns is varied from two meters up to sixteen meters with a step size of 2 m . By this, the influence of the size of the pattern shall be evaluated.

The movement speed of the UGV and the UAV is not examined during the simulations. This is the first approach to identifying the quality of the heuristic for the placement of the vehicles. In a later stage, the simulation can be improved to represent a more realistic movement of the vehicles with an acceleration phase, a movement phase, and a deceleration phase. The simplification is believed to affect the evaluated parameters likewise. Nevertheless, the simplification most likely improves the performance for movement along shorter distances, as the acceleration and deceleration phase is more relevant than for longer distances. Therefore, this inaccuracy is tolerated in order to perform the planned steps in the limited available time.

In the physics simulation, it is tested to transfer the concept of the Python simulation to software solutions as they are used with the actual hardware. The behaviour of the vehicles is investigated to get a better impression of the developed software before tests with the hardware are performed.

4.3. Field Tests

The field tests for the hardware demonstration are performed in a gravel pit in Planegg close to Oberpfaffenhofen which is used regularly for experiments by the DLR Institute of Communications and Navigation. It provides a rough terrain and at the same time an open space for flight experiments without endangering humans or risking damaging the property of the uninformed. As introduced before, the UGV is a Summit XL of the DLR Institute of Communication and Navigation. Its RTK-GNSS solution is ideal for a high positioning accuracy. As UAV, the presented Holybro S500 from the DLR Institute

of Flight Guidance is used. Its GPS-receiver for position determination has a poorer accuracy than the RTK system but it is still sufficient for the proof-of-concept. The sensor and gimbal are solely simulated due to the costs and availability of the TDLAS-sensor system. The setup of the field tests is shown in Fig. 4.13. To monitor the systems during the demonstration, an operator laptop with an integration of the ROS-network is used. To increase the monitoring capabilities, enable modifications, and ensure interoperability with the systems of the different institutes, the calculation node is executed on the operator's laptop. This part differs from the concept in which the node is executed on board of the UGV. This is an acceptable modification since the demonstration is not about communication solutions, but the cooperative positioning of the vehicles. The operator's laptop is located close but at a safe distance to the area, to ensure the safety of the operator in case of unforeseen behaviour or malfunctioning of the UAV or the UGV.

The base station of the RTK-system is located in the corner of the experimental area (blue). Its position is used as origin for the local reference systems of the vehicles. The x-direction of the vehicle's local coordinate system runs in the eastern direction, whilst the y-axis runs in the northern direction. The operator laptop is connected to the local network created by the network device (purple) at the UGV's base station.

To communicate with the UAV, the implemented radio solution in the 433 MHz band is used (green). To enable the laptop to send the commands, a USB to radio signal converter is used. Via this connection, messages of MAVLink protocol are exchanged with the UAV.

As in the Gazebo simulation, for the path finding of the UGV, the integrated path planner of the UGV superior to the rather simple A* algorithm is used. Therefore, a bitmap of the grayscale map is transferred to the processing unit of the UGV.

At the beginning of the demonstration, it is verified separately that the connections to the vehicles are available and the vehicles can be commanded by messages created at the operator station.

Afterwards, the demonstration is started and the TDLAS-manager node is commanding the vehicles. To ensure safety, a safety pilot is monitoring the UAV at every time to intervene in case of emergencies. Alike, a second operator is monitoring the UGV to intervene in case of malfunctioning to prevent the damaging of the hardware or injuring operators.

For the demonstration, the exploration pattern of the hourglass with a diameter of 8 m is chosen. It is best suited as it shows promising results in the simulation and results in good visual feedback for the operators.

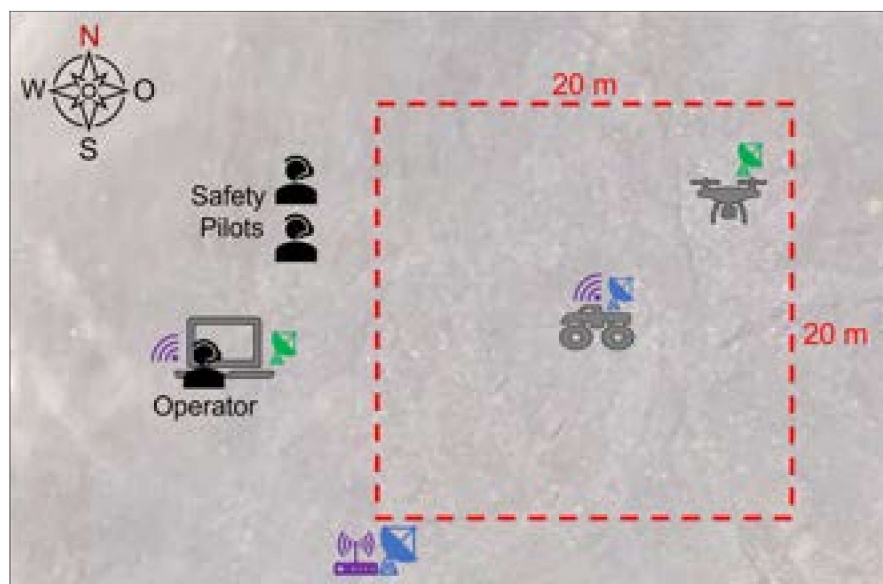


Figure 4.13.: The figure shows the setup for the proof-of-concept demonstration.

5. Results

At the beginning of this chapter, the results of the Python simulation are presented and discussed. More than 530 simulation runs are performed in the scope of this thesis. The most important findings from these simulations are presented in this section. Afterwards, the physics simulation is analysed. It is investigated to what extent the simulation can be used to perform experiments before the hardware integration is realised. At the end of the chapter, the findings of the field tests are reviewed concerning the feasibility of the developed concept.

5.1. Python Simulation Results

5.1.1. Regularisation Factor

In the first simulation runs, an investigation of the regularisation factor r is performed. To make a short recapitulation, the regularisation factor r is the parameter on the main diagonal of the regularisation matrix R (see Eq. (3.3)). It is used to enable the reconstruction of the plume, even while the measurement matrix M is underdetermined. An influence of

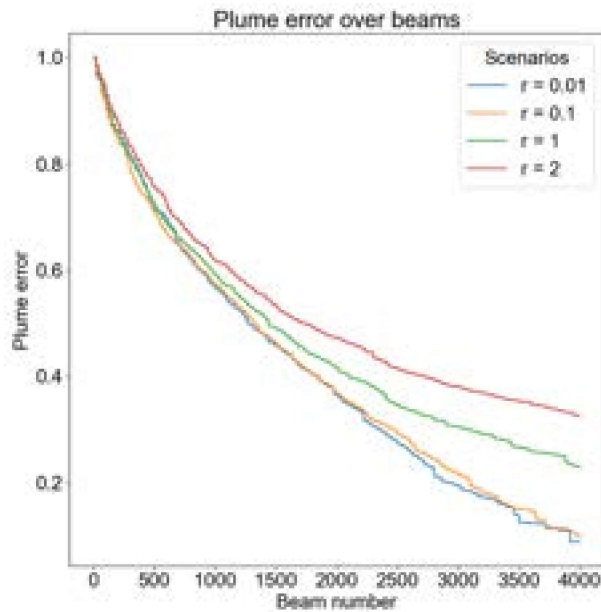


Figure 5.1.: The figure shows a plot of the simulation results in dependency of four different regularisation factors r . On the ordinate, the error of the reconstructed plume (4.2.4) over the number of measured beams on the x-axis is shown.

the scenario on the reconstruction process of the plume is not expected. For this reason, the investigation of r is performed in a single scenario only. For the executed simulation, the scenario with the empty environment and the cubic plume in the centre of the area is selected. To achieve a neutral evaluation, the simulations are performed with a random placing of the vehicles in the area. Four simulation runs with 4000 beams each are performed in which r is altered with the values 0.01, 0.1, 1, and 2. For the random placement of the measurements, the UGV is placed in a random voxel in the ground layer and the UAV in a random voxel with DLOS. As can be seen in Fig. 5.1, the variation of r in the four simulations leads to a difference in the reconstruction quality of the measured plume. As expected, a smaller value leads to a smaller remaining error after a fixed amount of measurements. With a higher value of r , the regularisation is rated higher. Hence, the estimated value for a voxel is distorted more. With a small value for r the regularisation is de-emphasised and the distortion reduced. Whilst the remaining error of the plume reconstruction with $r = 2$ is in the magnitude of 0.325, the remaining error with $r = 0.01$ is in the area of 0.089 for the 4000 beams. This leads to the conclusion, that a smaller value for r improves the achieved results. But it can already be seen between the simulation with $r = 0.1$ and $r = 0.01$ that the improvement stagnates. To confirm the trend an additional simulation run with a value of $r = 0.001$ was executed. That simulation run confirmed the trend.

However, in a second analysis the condition number of the reconstruction matrix for the simulations is investigated. A visualisation is shown in Fig. 5.2. In the figure, the condition number is plotted on a logarithmic scale over the beams measured during the

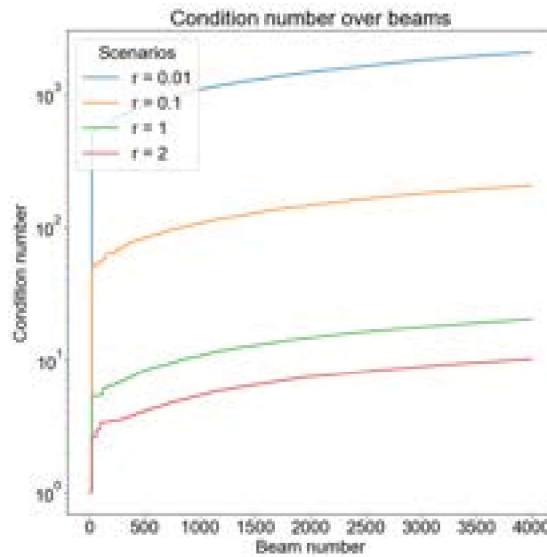


Figure 5.2.: The plot shown in this figure shows the development of the condition number of the reconstruction matrix during the four simulations on a logarithmic scale as a function of the reconstruction factor r .

simulation. While the condition number with $r = 2$ is in the magnitude of $5 \cdot 10^0$ and for $r = 1$ is in the magnitude of 10^1 , it drastically rises for $r = 0.1$ to 10^2 and for $r = 0.01$ even to the magnitude of 10^3 . With the condition number, the convergence of iterative methods and the influence of measuring errors is evaluated [27]. If the condition number is high, it takes longer to solve the equation system. Hence, higher condition number leads to an increased calculation time. This is not desirable because it increases the simulation time and endangers the real-time capability. Furthermore, with a high condition number inserted measuring errors have a greater impact [27]. Small changes can lead to big alterations which results in an instability. As a result, a low condition number is desirable. For these reasons, especially $r = 0.01$ but also $r = 0.1$ are not suited for the intended use.

This leads to the conclusion to perform the simulations with a regularisation factor of $r = 1$. It represents a balanced compromise between reconstruction capabilities and quality.

5.1.2. Random Approach

For each scenario introduced in Section 4.2.3 a simulation with 4000 measurements is performed. The results of these simulations are used as a reference of the performance for the heuristic. In the reference simulations, the vehicles are positioned randomly in the area without trying to optimise the placement in any manner as it was the case for the specification of the regularisation parameter.

The results show that the various surrounding configurations only have little influence on the resulting plume error. The divergence between the different surroundings for a single plume is similar. Hence, it is discussed exemplary on the centred plume. The plume errors of the five different surroundings with the centred plume are displayed in Fig. 5.3. The plot shows the plume error on the axis of ordinates plotted over the number of used beams on the abscissa. In the graph in Fig. 5.3, it is seen that the plume error is strictly decreasing for all five simulations with an increasing number of measured beams. After the 4000 measurements, the mean error of the five simulations is 0.227. The maximum difference is 0.024 between the error in M1-1 with an error of 0.238 and the error of M1-1 with 0.214. After the last measurement, the standard deviation is 0.0088. The moving distances of the UAV and the UGV on average show a linear increase during the simulations. This behaviour is expected. Throughout a single simulation, the distance between the successive target locations varies. Averaged over 4000 measurements a continuous increase is the outcome. This results in an average distance of 39.34 km for the UGV and 48.88 km for the UAV. These are extremely long routes. It is relativised by the average simulated time of 9.16 hours. This points out the importance of optimising the placement of the unmanned vehicles. Moreover, it demonstrates the necessity for long-term monitoring with the TDLAS technology and its impracticality for fast, high-resolution results. These conclusions indicate that the influence of the surroundings is small enough that findings of simulations in a single surroundings configuration can be transferred to the

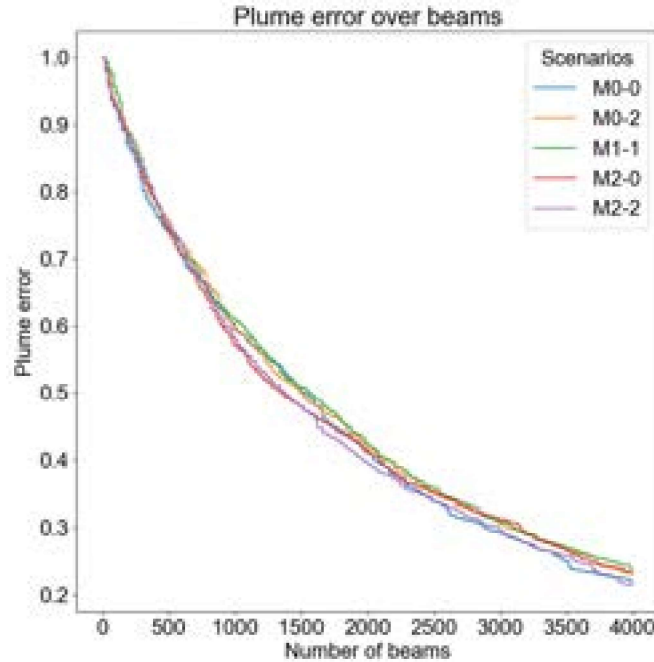


Figure 5.3.: In this figure, the measurements from 4000 beams in the five simulations in different surroundings are plotted. For a more detailed analysis, the plot is complemented with the mean and the standard deviation of the five simulations.

other four surroundings.

Another observation is made for the positioning of the plumes. In Fig. 5.4 five simulation runs with the different plumes in Mo-o are depicted. The graph shows the plume error on the ordinate over the number of beams on the axis of abscissas. For all five plumes, a strict decrease of the plume error with more beams is visible. But the error curve of the different runs is spread significantly wider after the 4000 measurements. The mean plume error of the five different plumes in Mo-o is 0.265. Regardless, the maximum difference of 0.092 is between the layered plume in the centre and the plume at the side of the map. This is nearly four times the maximum spread in the different surroundings configurations. The standard deviation at this point is 0.0404. Additionally, Fig. 5.4 shows that for measuring the layered and the centred plume, similar results are achieved. It is reasonable, that those two plumes are surveyed in more detail because random beams in the area of interest are more likely to cross the centre of the area than a corner. Furthermore, voxels in the centre of the area can be measured from more different angles than voxels located at the side of the area. The double and side plume retain the biggest plume error after the reconstruction. This is plausible based on their placement in the area. At first, surprisingly the performance for measuring the double layered plume is better than for the double plume. This the case for all five surrounding configurations. For unmeasured voxels, it is assumed that the concentration is 0. Thereby, the error for unmeasured voxels containing a high concentration is bigger than for voxels containing a low concentration.

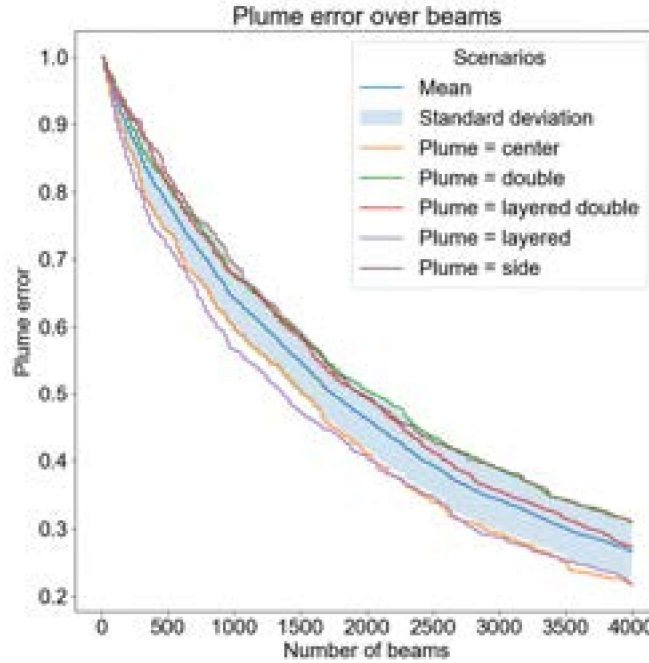


Figure 5.4.: The plot in this figure shows the plume error of five simulations with different plumes in the empty environment. For the measurements, the vehicles are placed randomly in the environment.

The double plume and the double-layered plume have the same dimensions but differ in the overall concentration. The outer layer of the double-layered plume only contains a concentration of 0.5. This results in a smaller plume error after the measurements. For this reason it makes sense at a second glance. The analysis shows that the course of the graphs for the different plumes is similar. They only vary in the rate of decrease.

The mean travelled distance of the UGV deviates just for a few 100 meters compared to the mean distance from the prior discussed simulation surroundings. The discrepancy for the travelled distance of the UAV is smaller than 100 *m*. These values are in the per mill range compared to the overall travelled distance. Also, the estimated time for the 4000 measurements deviates only in the range of a few minutes. This result is as expected since the process for the placement is not changed.

With the knowledge gained about the different scenarios and the reference values, the performance of the exploration concept is evaluated in the next sections. Besides, the analysis shows that the different surrounding configurations do not lead to strong discrepancies in measuring the plume. The different plumes alter the gradient of the error reduction with the number of measurements. Consequently, it is decided that the weighting parameters of the cost function can be determined with a selection of simulations. Once a set of parameters has been identified, it is verified with the whole set of all 25 scenarios.

5.1.3. Cost Function Parameters

In this subsection, the three weighting factors for the elements of the engineered heuristic presented in Eq. 3.4 are analysed. In the end, the weighting factor for the modified version of the heuristic is investigated. To apply the heuristic, a measuring pattern is required. The quality of the patterns is unknown until it is assessed with the heuristic. The hourglass and cross pattern as well as the inverted pyramid and the pyramid are alike and the limited time does not allow to perform the evaluation of the weighting factors with all patterns and configurations. Hence, it is decided to perform the evaluation with the hourglass with a size of 2 m by 2 m and the inverted pyramid with an edge length of 3 m by 3 m . For consistency between the different evaluation runs, the scenario in the map Mo-o with the centred plume is used for all parameter evaluations. In addition, the results are verified with further scenarios on a random base. Full coverage of all parameter variations with all scenarios is not possible due to the required simulation time in the given time frame.

Evaluation of d

For the first discussion, the weighting factor d of the distance term is varied. The other two weighting factors m and c are set to 1. To gain an insight into the impact of d , it is varied in a wide range between 10^{-5} and 10^4 . For the discussion of d , in Fig. 5.5 the plume

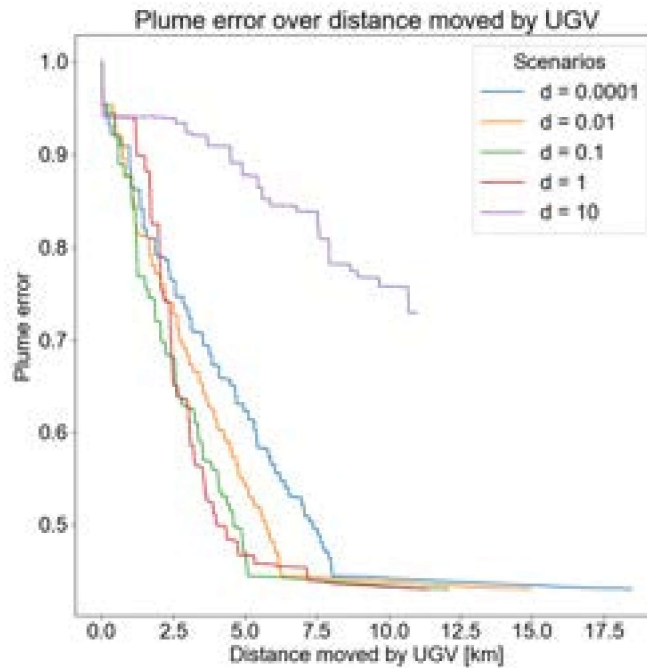


Figure 5.5.: The figure shows the evaluation of the weighting factor d based on the plume error. The plume error is plotted over the distance moved by the UGV. During each simulation, 6000 beams were created to measure the plume.

error for different values of d is plotted over the distance moved by the UGV. The shown simulations are performed in the Mo-o surroundings with the centred plume. Each simulation creates 6000 beams. In Fig. 5.5 it is visible, that the simulation with $d = 10$ is achieving significantly worse results than the other shown runs. After the 6000 runs, the plume error is still 0.73. The results for simulations with $d = 100$ and above perform worse. Further, it can be seen that the UGV has to travel a greater distance to achieve a reduction for the plume error compared to the other values for d . At the end of the simulation, the UGV travelled nearly 11 km. With decreasing values of d down to 0.1, the performance of the heuristic improves. The plume error is reduced to values around 0.43 and the UGV only moves around 5000 m to achieve this reduction in the error. Decreasing d further does not lead to a further significantly reduced plume error. However, the distance required to travel for the reduction is increased again. For $d = 0.01$ around 6000 m and for $d = 0.0001$ around 8000 m are driven by the UGV. Noteworthy is the stagnation of the plots after a certain plume error has been achieved. It marks the best reconstruction result that can be achieved with the used pattern for the exploration and mapping of the plume. This topic is discussed further later on in the section on the evaluation of the different measuring patterns.

To supplement the findings shown in Fig. 5.5, in Fig. 5.6 the distance travelled by the UGV is plotted over the time. An increased value of d causes the heuristic to avoid travelling long distances without performing measurements. The result is a reduction of the overall travelled distance. On the downside, widespread plumes are mapped rather poorly. The smaller the chosen value for d gets, the longer the overall travelled distance is. Due to the small size of the area and compared to the fast travelling speed as well as simplified

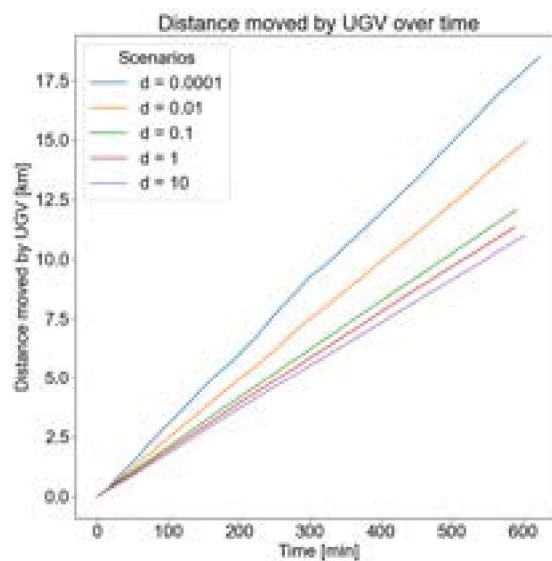


Figure 5.6.: In the figure the distance travelled by the UGV is plotted over the time for different values of d .

movement, the increased distance does not lead to a significant increase in measuring time. With $d = 0.0001$ the UGV travels 18.46 km whilst for $d = 10$ it only moves 10.99 km . This is an increase of the distance by around 60 % with a time difference of around 16 minutes. Similar observations are also made for other simulations performed with the hourglass pattern with the Mo-o surroundings and the layered plume, the hourglass pattern in the M1-1 surroundings with the centred plume, as well as the evaluation with the inverted pyramid pattern. To investigate the envisaged range around $d = 0.1$, five random samples are performed between 0.01 and 10. The results verify the observed trend. To achieve the optimal performance of the heuristic, a more detailed analysis has to be performed. As the objective of this thesis is the proof of concept and not its optimisation, the investigation is stopped with this result for d . Therefore, the presented level of detail is sufficient. With the knowledge gained from this analysis, it is determined to set d to 0.1.

Evaluation of c

In the second step, the weighting factor c for the concentration addend of the heuristic is examined. For this simulation step, the other two weighting factors m and d are set to 1. For the evaluation of c , the weighting factor is modified in the range between 10^{-3} and 10^5 with a step size of 10^1 . Additional, simulations to narrow down the decision are executed where needed. In Fig. 5.7 a selection of the results for simulations with the hourglass and inverted pyramid pattern are depicted. The plume error is plotted over the time for both measuring patterns. Since the UGV does not move during the inverted pyramid measuring pattern, it is not reasonable to plot its plume error over the distance travelled by the UGV. The simulations plotted in Fig. 5.7 are performed in the Mo-o surroundings and the centred plume. Each simulation performs measurements with 6000 beams. In the two plots depicted in Fig. 5.7, it can be seen that an increase of c leads to a faster reduction of the plume error. It is faster in regard to time as well as for the travelled distance of the UGV. Additionally, the course for the plot of $c = 10000$ shows that the further increase c leads to a worsening of the result. For $c = 10000$ the plume error with the hourglass pattern reaches the stagnation point after approximately 6000 m and after approximately 220 min with the inverted pyramid pattern. Simulations with higher values up to $c = 10^5$ confirm this trend. For this value, the reduction of the plume error is slowed down, and it does not reach the lowest possible error with the pattern during the 6000 measurements. In the analysis with the widespread range, the best results were achieved with $c = 1000$. Therefore, for the narrow analysis of c , the focus is on that magnitude. The large steps for the value of c compared to the analysis of d , create a rather small impact on the performance of the heuristic. In the comparison of $c = 1000$ and $c = 1500$ just small improvements are noticeable. For the hourglass pattern $c = 1000$ stagnates after approximately 2300 m and for $c = 1500$ after 2000 m . For the time passed with the inverted pyramid, it is 118 minutes for $c = 1000$ and 93 minutes for $c = 1500$.

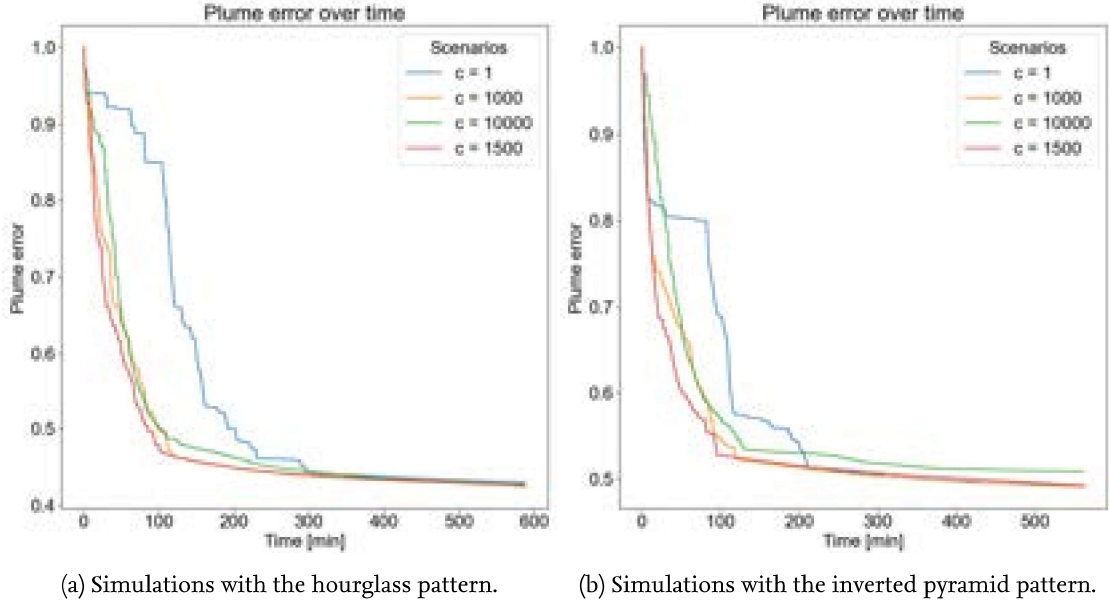


Figure 5.7.: The left side of the figure shows the plume error over the time for four simulations with the hourglass pattern (a). The right side shows the plume error plotted over the estimated required time for measurements performed with the inverted pyramid pattern in four simulations (b).

However, it can be seen that the simulations with the hourglass pattern stagnate around a value of 0.45 whilst the simulations with the inverted pyramid pattern already stagnate at a value of 0.51.

The presented results were additionally verified with simulations of the hourglass performed in M1-1 with the centred plume and in Mo-o the layered plume. The analysis of the travelled distance for the vehicles does not show a wide range and is similar in all investigated configurations. Hence, it can be concluded that the variation of the concentration weighting factor c does not have a significant impact on the travelled distance. The findings of the analysis lead to a choice of $c = 1500$ for further simulations.

Evaluation of m

In the next step, an analysis of the weighting factor m is performed. For this examination, the weighting factors d and c are set to 1. The analysis of m is executed in the range of 10^{-3} to 10^5 . Fig. 5.8 shows multiple simulations performed with the hourglass pattern in Mo-o containing the centred plume. On the axis of ordinates, the plume error is plotted over the distance moved by the UGV displayed on the abscissas. Also in this configuration, 6000 beams were performed for each simulation run. With $m = 0.1$ only a bad performance of the heuristic could be achieved, as the plot shows. It takes a long time to reduce the error and the UGV to move for a far distance. Increasing the value for m leads fast to an improvement of the performance. A narrow evaluation for values of m between 0.1 and 2

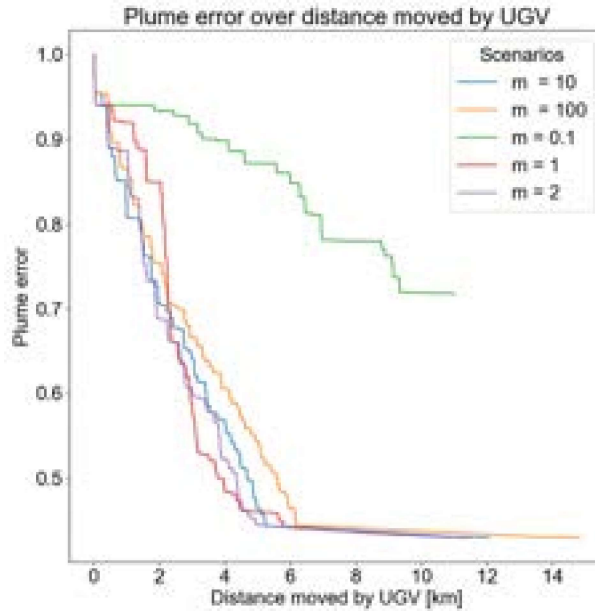
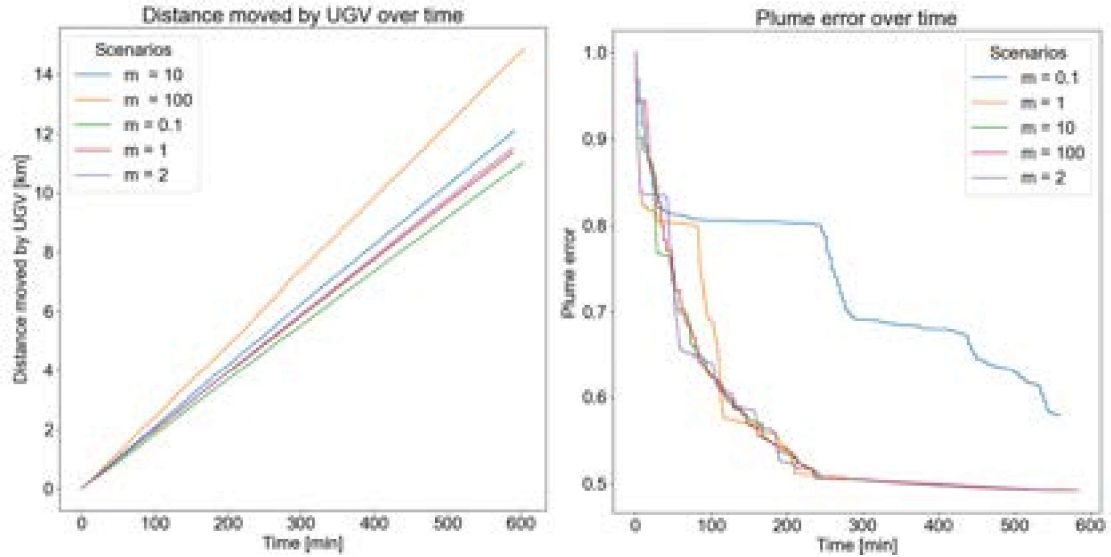


Figure 5.8.: The figure shows the plume error over the distance moved by the UGV for simulations with the hourglass pattern. Plots for multiple values of m are displayed.

confirmed the trend. Increasing the value further to 5 and above leads again to a decrease in performance. This can be seen in the plot of $m = 10$ and $m = 100$. The narrow evaluation was reinforced by simulations in the M1-1 environment with the centred plume. However, for higher values, the rate of deterioration is reduced and stagnates. Simulations with values higher than $m = 100$ do not attain significantly worse results. In Fig. 5.8 it can be seen that the value of m does not alter the minimum achievable plume error with the pattern, but only the time it is obtained in. As stated in the definition of the heuristic, the measurement addend is used to create an exploration effect. It is supposed to stimulate the vehicles to move away from locations that already have been measured in detail. In Fig. 5.9(a) the distance travelled by the UGV is displayed over the time. The expectation that a higher value for m leads to more movement is confirmed. Further measurements have shown that for values of $m > 100$, the increase in travel distance stagnates. For that reason, they are not shown in the diagram. This is coherent with the stagnation of the plume error performance for higher values of m . The reason for this behaviour is the size of the examined area for the simulations. At this point it is no longer possible to move away faster. Therefore, a further increase of m has no influence anymore. Nevertheless, a shorter travelling distance is desirable. For this reason, at the current time, $m = 2$ is seen as the best choice. It creates a fast reduction of the plume error and a good performance with regard to the travelled distance.

The same analysis is also conducted with the inverted pyramid pattern. The results are depicted in Fig. 5.9(b). It also states that small values for m slow down the rate of error reduction. Nonetheless, a difference compared to the simulations with the hourglass



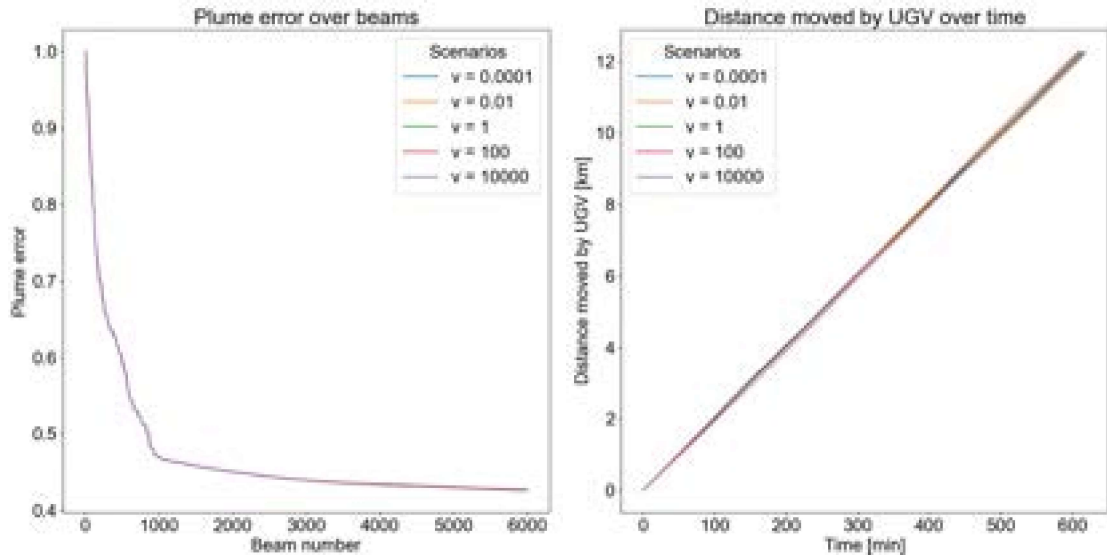
(a) The plot shows the distance moved by the UGV during simulations with the hourglass pattern over time. (b) The figure depicts additional plots of simulations performed with the inverted pyramid pattern to evaluate m . The plume error during the simulations is plotted over time.

Figure 5.9.: The two plots shown in this figure substantiate the evaluation of the weighting factor m .

pattern is visible. Higher values for m do not result in a noticeable deceleration of the error reduction. All values for m reach the stagnation point after around 230 min. An exception is the outlier $m = 0.1$. Since the simulations performed with the inverted pyramid do not contradict the prior findings, the value of m is set to 2 for further examinations.

Evaluation of v

In the last parameter analysis, it is investigated if the performance of the heuristic is improved by adding the variance term presented in Eq. (3.5). The other three weighting factors c , m , and d are set to 1 again for this investigation. For the variation of v simulations between 10^{-4} and 10^4 , are performed. Due to the required time to calculate the variance matrix to determine the variance of each voxel V_i , the runtime of the simulations is drastically longer than for the prior performed simulations. For this reason, the simulations are only performed with the hourglass pattern. Regardless, for each simulation 6000 measurements are carried out. In Fig. 5.10 a selection of the results for the v evaluation is presented. In the left half of the figure, the impact of different values for v on the reduction of the plume error is displayed. It is apparent that the alteration of v does not have any noticeable effect on the plume error reduction. In the right part of the figure, the distance moved by the UGV is plotted over the time. Here as well it is clearly visible that there is no influence in the travelled distance beyond the noise.



(a) Plume error over the beam number for different values of v . (b) Distance travelled by the UGV over the time for different values of v .

Figure 5.10.: The left plot shows the plume error over the beam number for different values of v (a). The right plot shows the distance travelled by the UGV over the time for different values of v (b).

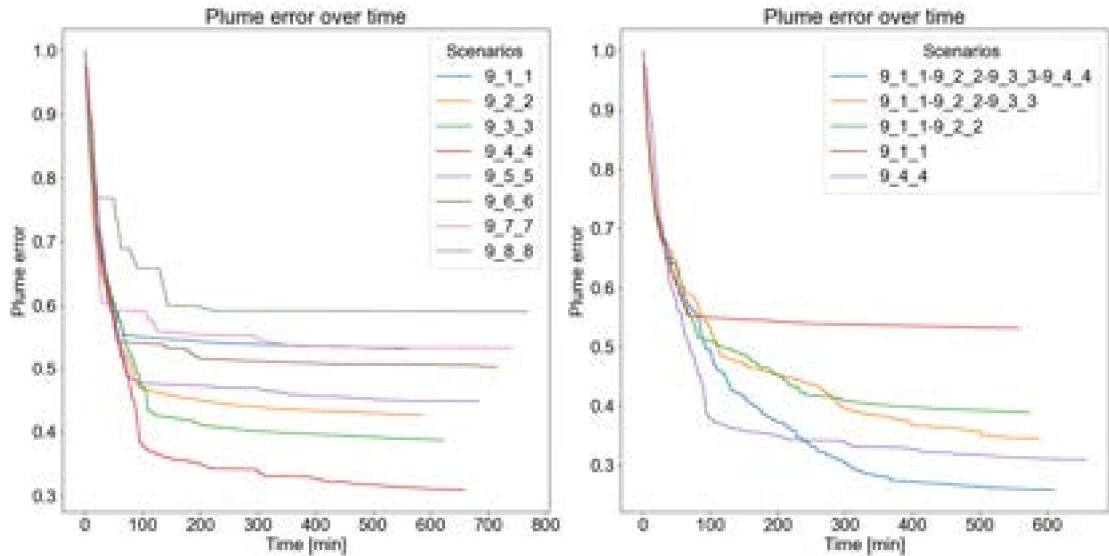
Concluding, the utilisation of the variance term in the heuristic leads to a drastic increase in calculation time for identifying target locations. In return, no noticeable improvement can be perceived. Consequently, the variance term is not examined further. In the subsequent simulations, the heuristic is used as introduced in Eq. (3.4). At this point should be mentioned that the implementation of the heuristic is normalised with m . Thereby, the direct dependency of d/m and c/m is created. To reflect the dependency in the parameters, d is set to 0.2 and c is set to 3000, and m is set to 2 in the simulation configuration.

5.1.4. Pattern of Interest

For all simulations discussed in this subsection, the prior analysed values are used for the weighting factors. In this section, the different patterns and a variation in their configuration are examined and compared in regard to their performance with the developed heuristic. The analysis for each pattern contains a discussion of different variations in the size of the pattern. The maximal radius of each pattern is varied from one to eight meters. The patterns are named by the following scheme: "*pattern name*"_"*height*"_"*y-radius*"_"*x-radius*". Also, an examination with a combination of the same pattern in different sizes is performed. It is tested how the performance changes when the algorithm is allowed to choose from multiple patterns of the same kind with a different radius.

Hourglass

For a start, the hourglass pattern used for the prior parameter evaluations is examined in detail. With an hourglass pattern of a single size, the pattern performs measurements from nine different angles. For each variation with different pattern sizes, eight further possible angles are added. The ninth angle is the vertical measurement which is identical for all configurations. The simulations shown in the plots in Fig. 5.11 are performed in Mo-o with the centred plume. On the left side of the figure in (a) the size of the hourglass is modified. The smallest hourglass has a one-wide ring around the centre. This results in a square of 3×3 voxels. With this configuration, the plume error can be reduced to 0.55. By increasing the size of the pattern, more voxels are measured with a single beam, since the opening angle of the hourglass shape is increased. Thereby, the plume error can be reduced further to nearly 0.3 with an hourglass with a ring of four voxels around the centre. Increasing the size of the pattern further up to eight voxels around the centre impairs the achieved plume error. With a distance of seven voxels to the centre, the results are similar to the one-wide ring. The hourglass with an eight voxel wide ring achieves even worse results. This is traced back to the size of the area. Due to the form of the hourglass, a cutout close to the border of the area is created that can only partially be measured by the pattern. Near the border only beams running parallel to the border can pass through the voxels. Beams perpendicular to the border are not created, since it would require one of the vehicles to be outside of the area. With the growing size of the pattern, the cutout is increased. At a certain point, the reduced number of measurements in the cutout area leads to a reduction in the mapping quality. Another aspect visible in Fig. 5.11 (a) is the difference in the expected measuring time. A smaller pattern size results in a shorter time. This fits to the travelled distance. Larger patterns lead to an increased travelling distance for the UGV and the UAV. This is plausible since the vehicles have to travel further between the measurements. In addition, another analysis of the result shows that in general, the UAV is moving further than the UGV. At first sight, the result seems bizarre because the two vehicles are moving in the same pattern. On second thought it is correct as explained in the following. To determine the order of the measuring points for the next pattern, the travelling salesperson problem is solved. This includes the shortest path from the current position to the first point of the measuring pattern. Since the problem is solved for the UGV, the point closest to the position of the UGV is chosen. This means that the UAV has to move from the far side of the current pattern to the far side of the next pattern. Summed over all the patterns performed during a simulation, it leads to a perceptibly increased distance. It can be said that this is inconsistent because the UAV has a shorter battery life than the UGV. But the UAV also consumes battery power while it is waiting for the UGV to arrive at its target. To optimise this aspect, it is suggested to perform a trade-off between the increased battery consumption for the UAV during movement compared to hovering and the travelling time for the UGV. In Fig. 5.11 (b) it is shown how the performance changes when the algorithm is allowed to choose between different patterns. Due to the



(a) The figure shows a plot of the plume error over time for different sizes of hourglass patterns (b) The figure displays the plume error over the time for multiple simulations. In each simulation the algorithm can choose from different hourglass patterns.

Figure 5.11.: Evaluation results for the hourglass pattern

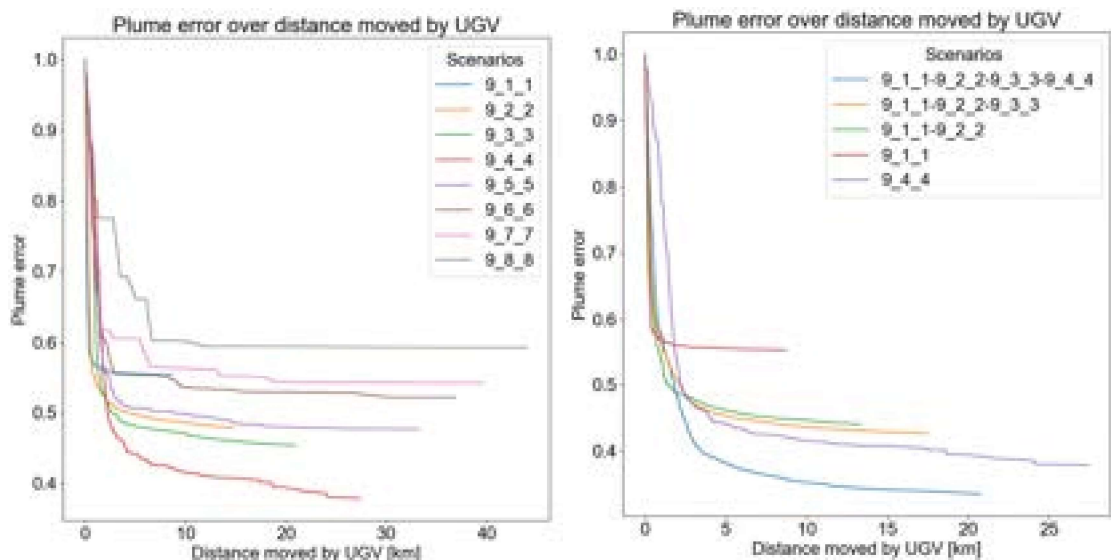
limited resources for the simulations, it is only examined up to four different patterns. The ability to choose from different patterns, also increases the calculation time. The algorithm has to calculate the value for every possible placement of each pattern. Thereby, by doubling the patterns, the calculation time is doubled. In the plot, it is visible that the combination of the patterns leads to a reduction of the minimum achievable plume error. The pattern 9_4_4 performs best in the single configuration evaluation with a plume error of 0.3. The combination of the four smallest patterns leads to a plume error of approximately 0.27. Though, it takes a longer time and more measurements are required to reduce the plume error than with the single pattern. Examining the distance travelled by the vehicles, it is determined that the UGV travels 23.67 *km* for the simulation with the 9_4_4 pattern and only 16.12 *km* during the simulation with the four small patterns. A similar difference is noted for the UAV. This means that the choice of pattern leads to a reduction of plume error as well as a significant reduction in travelling distance.

As clarified at the beginning, a single version of this pattern contains nine different beams. Thereby, nine measuring angles are applied to measure the gas concentration in the voxels. In the scope of this thesis, a measuring angle is the orientation compared to the crossed voxels in the area. The orientation in all three dimensions has to be taken into account. Different-sized patterns lead to a variation in the measurement angles. For each additional version of the hourglass pattern, eight further measuring angles are possible. Considering a single voxel, this also means that the beam crosses different neighbouring voxels depending on the measuring angle and thereby another value is measured.

Concluding, it introduces a new independent line to the measuring matrix. As the plots show, a single pattern can only be applied to an area a limited number of times. Once this point has been reached, the improvement of the plume error stagnates. By adding further patterns, more independent rows are appended to the measuring matrix. Hence, a better reconstruction of the plume is performed. The takeaway from this analysis is that the selection from the different patterns and thereby the increased number of possible measuring angles is desirable.

Cross

The findings from the hourglass pattern are confirmed by the simulations with the cross pattern. It is an interesting follow-up since it is a simplified version with only five angles for each pattern instead of the previous nine.



(a) The plot displays the plume error over the distance moved by the UGV for different sizes of the cross pattern. (b) The graph depicts the plume error over the distance moved by the UGV for simulation runs with a choice of multiple cross patterns.

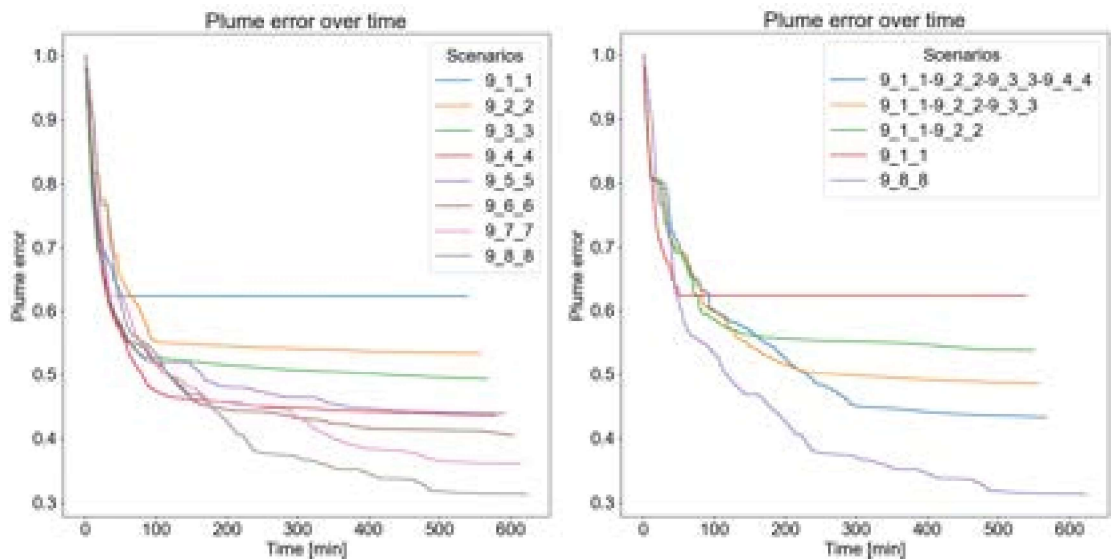
Figure 5.12.: Evaluation results for the cross pattern

For comparability, the simulations are also performed in Mo-o with the centred plume. In Fig. 5.12 the results for the evaluation of the cross pattern are displayed. On the left side in Fig. 5.12 (a) the various sizes of the cross pattern are examined. It can be seen that up to 9_4_4 the plume error can be reduced to 0.38. This is not as good as with the hourglass pattern. The reduced performance is lead back to the lower number of available measuring angles. Increasing the pattern further leads to a deterioration of the plume error. Since the plot is showing the plume error on the axis of ordinates over the distance travelled by the UGV on the abscissa, the different travelling distance is visible. Again, a smaller pattern size results in a smaller overall travelling distance. In Fig. 5.12 (b) the same observations

as for the hourglass pattern can be made. If the algorithm can choose from multiple patterns, it reduces the plume error further than if only one of these patterns would have been used. But it is not reduced further, than a better performing pattern. With the cross 9_1_1, 9_2_2, and 9_3_3 the results are not better than pattern 9_4_4. Adding cross 9_4_4 to the list leads to a performance exceeding the single cross 9_4_4. The best performance achieved with the cross pattern therewith is a plume error of 0.36.

Inverted Pyramid

Next, the inverted pyramid pattern is examined. It involves four different measuring angles for each application. The simulations are as well performed in the scenario with a centred plume and the surroundings Mo-o. In Fig. 5.13 the simulation results for different sizes of the pyramid are compared. The plume error on the ordinate is plotted over the time on the abscissas. The plot shows that increasing the edge length of the pyramid leads



(a) The figure shows plots for simulations with different sized inverted pyramids. It shows the plume error over the time.
(b) The plot shows the results of simulations with different combinations of inverted pyramid patterns. It shows the time on the axis of abscissas and the plume error on the ordinate.

Figure 5.13.: Evaluation results for the inverted pyramid pattern

to an improved reduction of the plume error. In contrast to the prior examined patterns, the trend continues up to the 9_8_8 pattern to a plume error of 0.31. The course of the graphs is similar for the different sizes of the inverted pyramid. In the beginning, the plume error is reduced fast. At a certain point, the plume error reduction is slowed down until it stagnates towards the end. It stands out that large configurations of the pattern reduce the error with a similar rate as the smaller versions of the pattern. The flattening of the plume error curve increases after the curve for the smaller pattern already began to

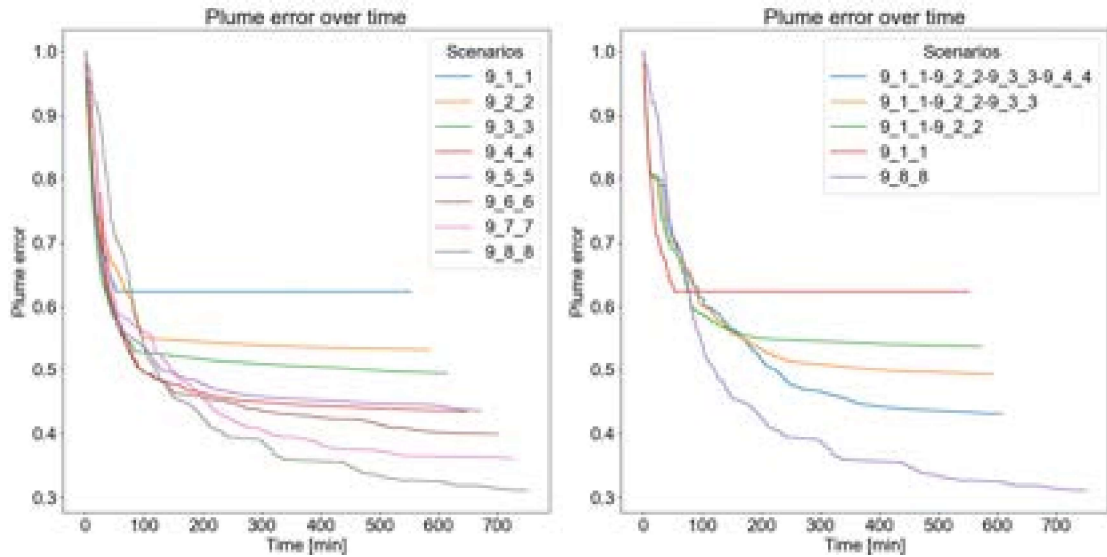
flatten or even stagnated. Since the UGV only moves between the patterns, it moves for a rather short distance. For the different sizes, it varies between 1800 *m* and 2500 *m*. A clear dependency on the size cannot be identified. This is different for the movement of the UAV. Here it can be said clearly that the size of the pattern defines the moving distance. With the increase of the inverted pyramid size, the distance travelled by the UAV rises. Whilst it moves less than 10 *km* for the 9_1_1 pattern, it travels nearly 60 *km* for the 9_8_8 pattern.

On the graph shown in Fig. 5.13 (b), the inverted pyramid is examined for different combinations of sizes. Once more, it can be observed how the combination of different pattern sizes improves the performance of the target identification algorithm. The more patterns are combined, the better is the result. Still, the biggest pattern of 9_8_8 performs better than the combined pattern. A simulation with a combination of multiple large inverted pyramids has not been performed. Therefore, this thesis leaves a gap at this point to be investigated by follow-up studies. Based on the performance of the single 9_8_8 inverted pyramid, a combination of large pyramids may exceed the performance of the 9_4_4 hourglass. However, the inspection of the travelled distances points out the disadvantage of this pattern. Whilst the combination of the patterns reduces the movement of the UAV, the movement for the 9_8_8 inverted pyramid is more than double the combinations of the four patterns 9_1_1, 9_2_2, 9_3_3, and 9_4_4. Concluding, the UAV moves more than triple the distance compared with the 9_4_4 hourglass. Due to the limited battery power, this is a strong disadvantage for the pattern.

A later study should investigate the effect of combining the larger patterns. Because of its load to the UAV, the pattern is unattractive. Due to time constraints and comparability, it is not performed in the scope of this thesis.

Pyramid

The pyramid is similar to the inverted pyramid and so are the results. But for this pattern, the UAV mostly has to hover. On the downside, due to the terrain and the reduced movement capabilities of the UGV, the execution of the pattern requires more time. The simulations are performed in the same scenario as the other patterns. The results of the simulations for the pyramid are depicted in Fig. 5.14. On the left, the pyramid size is examined. The plume error is displayed on the axis of ordinates over the time on the axis of abscissas. On the right side, the combination of different pyramids is analysed. The results seem identical to the inverted pyramid. However, if one considers the time, it can be seen that significantly more time passes until the results are achieved. This is the expected outcome due to the prolonged travelling time of the UGV. The longer distances to be moved for bigger pyramids, like the 9_8_8, increase the problem. For these reasons, also the pyramid is seen as a problematic pattern for the measurements. Overall it provides too few measuring angles for a high travelling distance and time.



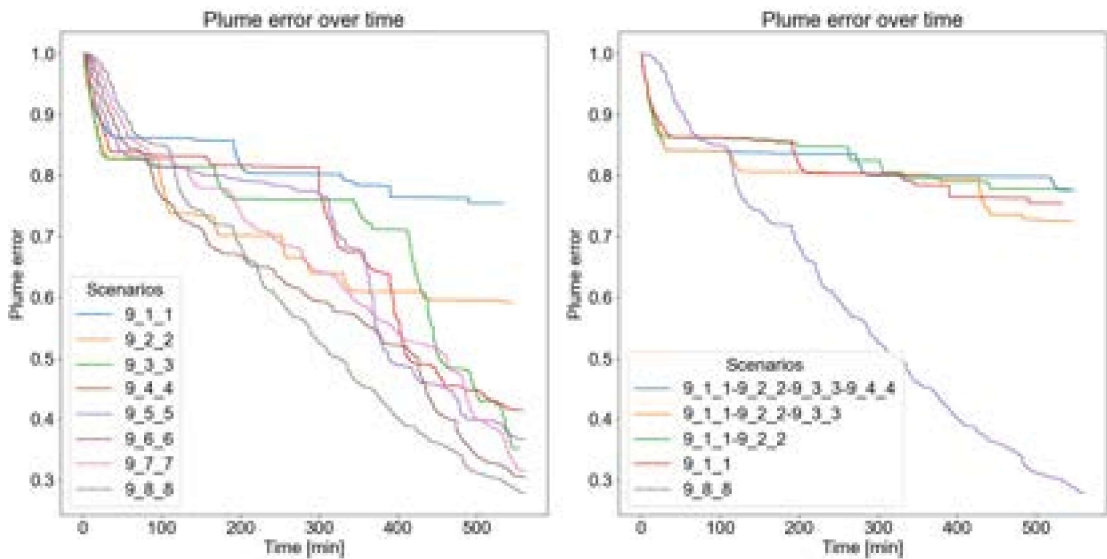
(a) In the figure plots for simulations with different sizes of pyramids are depicted, showing the plume error over the time. (b) In the plot the results of simulations with different combinations of pyramid patterns are presented. The time is plotted on the axis of abscissas and the plume error on the axis of ordinates.

Figure 5.14.: Evaluation results for the pyramid pattern

Cube

The last pattern evaluated is the cube. The number of measurements performed with the pattern depends on its size. Whilst with the 9_1_1 pattern 73 measurements are performed, the largest investigated pattern 9_8_8 creates 319 measurements for a single application of the pattern. All these measurements provide separate measurement angles. The simulations displayed in Fig. 5.15 are performed in Mo-o with the centred plume. The examination of the different-sized cubes in Fig. 5.15 (a) points out that the plume error reduction occurs at a slower rate than for the prior investigated patterns. None of the considered configurations of the cube pattern reaches its stagnation point in 6000 beams. However, it can be said that smaller cube patterns reduce the plume error at a slower rate than larger versions of the pattern. The smallest pattern 9_1_1 still has a plume error of 0.75 after 532 *min* and 6000 beams. In contrast, the biggest version of the pattern 9_8_8 reduces the error to 0.28 in 560 *min*. It is in the magnitude of the lowest error achieved but takes more time. Reflecting upon the distances travelled by the vehicles, a strong inequality between the vehicles can be seen. For the bigger versions of the pattern, the UGV travels less than for the smaller ones. But still, during the smallest cube pattern, the UGV only travelled for 300 *m*. The UAV on the other side, passed a distance of 10 *km* for the small cube. In the simulation with the big cube, the UAV moves 30 *km*. The comparison of the different sized cubes shows another problem. Although the cube pattern provides many different measuring angles in a single application, it is relatively inactive. A mul-

titude of measurements is performed before the situation is reassessed and the vehicles move. This rather leads to exploitation than exploration. The bigger versions of the pattern compensate for this fact since they are relatively large compared to the area. The 9_8_8 basically already covers more than half of the area. Thereby, it could be compared to a static sensor in the area. To investigate this behaviour further, it is interesting to see the change in performance in an enlarged area. These examinations are recommended for follow-up studies. In Fig. 5.15 (b) multiple cubes up to 9_4_4 are combined. Here it is



(a) In the figure the results of differently sized cube patterns are plotted. (b) In the plot simulations with different combinations of cube patterns are shown. The plume error is plotted over the time.

Figure 5.15.: Evaluation results for the cube pattern

interesting to see that the combination of patterns does not necessarily lead to improving the performance. The combined patterns perform in the magnitude of the 9_1_1 pattern. This means that it is more efficient to use the larger patterns separately than to have the choice between a combination of them. It seems that the smaller ones are dominant during the selection process and thereby chosen more often. Hence, the larger patterns performing better are applied infrequently and cannot exploit their advantage. A statement about the reduction of the plume error before stagnation cannot be made since the reduction process is so slow and never reaches the stagnation point with the cube pattern. The movement of the UAV reinforces the prior described selection process. If the patterns are combined, the distance travelled is reduced in comparison to the biggest pattern in the combination. Still, the UAV moves further than in the smallest pattern in the combination.

To summarise, the cube pattern shows the potential to achieve a low plume error due to its variety of measuring angles. However, it is rather inflexible due to the number of measurements performed in a single application. In addition, the pattern relies on the

extensive movement of the UAV. Depending on the size of the pattern compared to the area, the cube equals a static sensor placed in a suitable spot.

Comparison of Patterns

After the individual discussion of the patterns, the performance of the different patterns is compared directly. Therefore, the best performing single configurations of all patterns are juxtaposed in the plot shown in Fig. 5.16. It shows the plume error of the patterns on

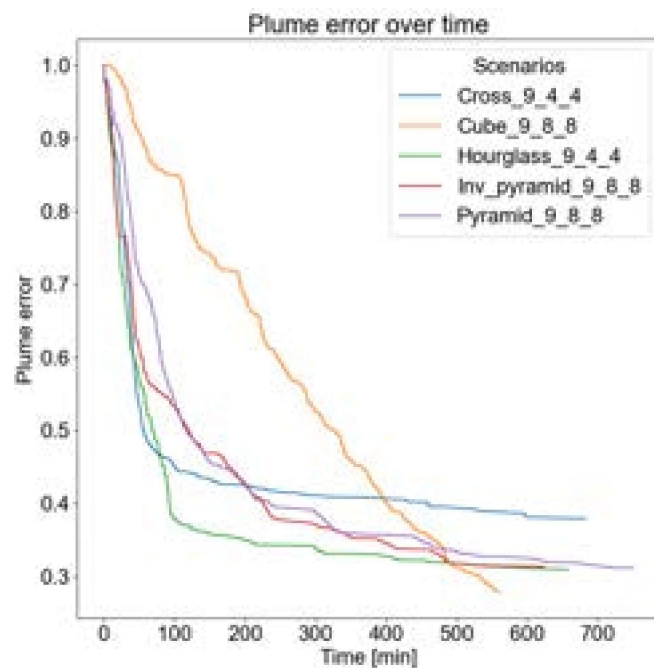
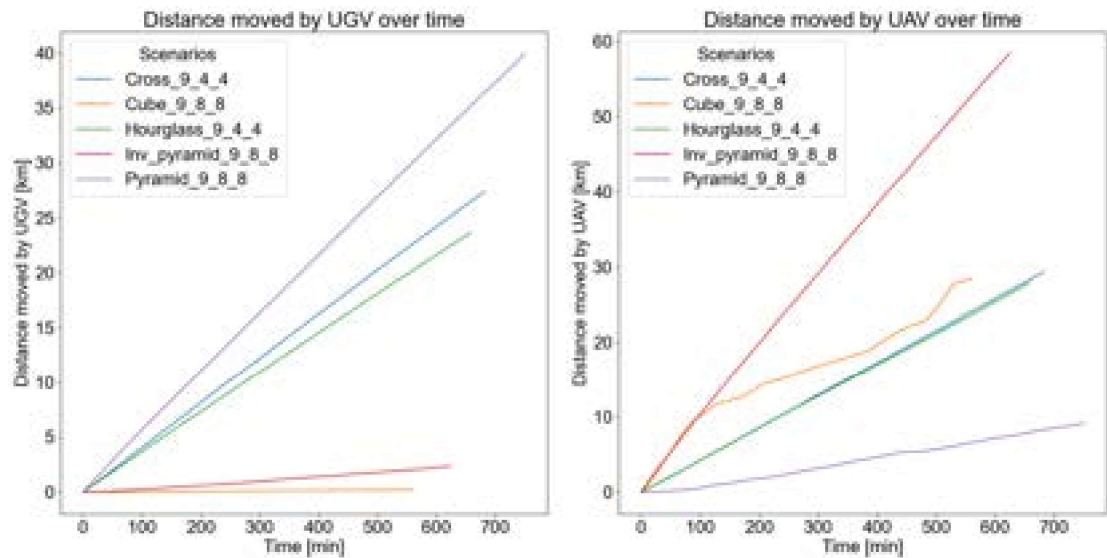


Figure 5.16.: The figure shows a comparison of the different patterns with the best performing configuration for each pattern.

the ordinate over the time on the axis of abscissas. The figure shows the prior addressed similarities between the hourglass and the cross. The course of the plots for the two patterns is similar. The cross stagnates earlier because it applies fewer different measuring angles. Prior to the reduction of the rate for the plume error is nearly identical. More similar is the behaviour of the pyramid and its inverted form. This is expected since the shape of the plumes is symmetrical about the height. If this was not the case, the pyramid would achieve better results for distributions closer to the ground whilst the inverted pyramid obtains a better resolution in altitudes below the UAV. Whilst for the cross and the hourglass the 9_4_4 configuration performs best, for the other three patterns the maximum investigated size of 9_8_8 achieved the best results. The cube 9_8_8 achieves the lowest plume error after 6000 performed measurements. However, the hourglass pattern requires only 106 *min* to lower the plume error to 0.37. The cube pattern needs more than four times the time to reduce the plume error to the same magnitude.

Additionally, the distances travelled by the vehicles have to be taken into account. In Fig. 5.17 (a) the distances for the UGV are depicted and in Fig. 5.17 (b) the movement of the UAV is visualised. Since the UGV does not move during the inverted pyramid and the cube pattern, the distance travelled by the UGV compared to the other patterns is short. As a result, the distance travelled by the UAV for the inverted pyramid exceeds the other patterns. The UAV has to travel nearly double the distance in contrast to the second. In comparison, the distance difference between the cube and the hourglass is small. But it has to be taken into account that the hourglass gains information in approximately 110 *min*. In this period of time, the UAV moves less than 10 *km*. That is less than a third compared to the 30 *km* for the cube pattern. The same is applicable to the cross pattern. This represents a large gap with regard to the travelled distance. The moving behaviour of the pyramid projects the opposite of the inverted pyramid. Therefore, it also shows a huge imbalance between the UGV and UAV. The distance travelled by the UGV exceeds the other patterns.



(a) Comparison of the distance moved by the UGV for different patterns. (b) Comparison of the distance moved by the UAV for different patterns.

Figure 5.17: The figure shows the distance travelled by the vehicles for the comparison of the different measuring patterns.

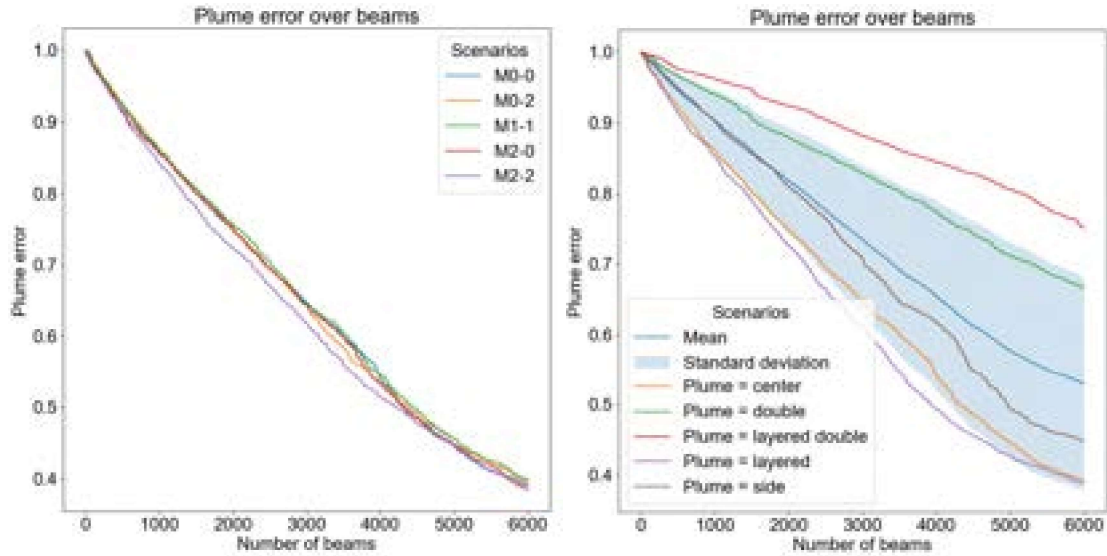
Additionally, simulations have been performed in which the algorithm for the target selection was allowed to choose from different patterns. The results show that the algorithm often chooses the cube which leads to a slow rate of plume error reduction. A simulation without the cube pattern performs better but falls short compared to the simulations only containing the hourglass pattern. This is traced back to the fact that the algorithm does not take into account from which direction a voxel has already been measured. It only sees how often the voxel has already been inspected. Hence, if the algorithm chooses one

of the patterns performing worse than the hourglass to investigate a set of voxels, it is unlikely that the voxels are investigated with the hourglass later on. Thereby, the resulting performance is worse than applying only the hourglass pattern.

Based on the analysis, a series of conclusions is drawn. It is desirable to have a pattern with multiple independent measuring angles. At the same time, the pattern must not contain too many beams so that the re-evaluation process is not delayed for long. If not enough measuring angles are available in the pattern, it is not possible to obtain a precise measurement of the present plume. Consequently, it is desirable to be able to choose from patterns containing different measuring angles. Though, the number of patterns must be limited. The number of possible patterns directly influences the calculation time. In addition, the presented heuristic does not take into account from which angles the single voxels have been measured. A modification of this sort will improve the performance for comparing multiple patterns. In the current configuration, voxels measured with a single pattern are measured less likely by another pattern, even if the other pattern uses a different set of measuring angles. With this measurement, additional information on these voxels is gained. Hence, the possibility to distinguish between the measuring angles is desirable. In the implementation, it can be realised by expanding the measurement addend of the heuristic with a series of look-up tables. In these, the angles are stored or in a simplified version angular ranges. If the investigated angle is new, the value of M_i is reduced and thereby more likely to be chosen.

5.1.5. Best Beam

An attempt to forgo the approach of the measuring patterns is performed with the best beam method. In this case, the beam with the lowest cost is calculated. This results in the conduction of a reconstruction after each measurement and hence in an increased processing time. A modification can be the determination of the next n-best beams to reduce the reconstruction effort. In the following, the results of the first attempt with the approach are discussed. For the best beam approach, the simulations were performed with all 25 scenarios. The rise in required processing time is severe. The prior presented simulation runs did not exceed an hour for each run. In contrast, the simulations for the best beam approach took more than seven hours each. It is traced back to the target acquisition process. With the size of the area, even with the limitations to the top and bottom layer, the amount of possible beams is too high for a brute force approach without sufficient computing power. In Fig. 5.18, a similar analysis as for the random beam is performed. Since the results are similar for the different scenarios, exemplary one analysis is performed for comparing the surroundings and another analysis to show the discrepancies for the different plumes. In Fig. 5.18 (a) the results for the centred plume in the different surroundings is presented. The results line out that the best beam method is not influenced significantly by the surroundings. This is reflected in the standard deviation as well. After the 6000 measurements, the best beam approach results in a plume error below 0.4



(a) The figure shows the comparison of simulations (b) The graph shows simulation runs in the Mo-o surroundings with the centred plume and the five different surroundings with the five different plumes.

Figure 5.18.: Results of the best beam evaluation

independent of the surroundings. The course of the graphs is almost linear. A reflection of the travelling distances displays a similar small spread. It stands out positively that the UGV only has to travel for around 8000 *m* to achieve the reduction of the plume error. The UAV covers a distance of 25 *km*. Although the differences for the various surroundings in regard to the travelled distance are not significant, it is visible in the results that the vehicles have to take more detours with an increase in the obstacle number. To perform the 6000 measurements the vehicle requires a little less than 11 hours.

The plot in Fig. 5.18 (b) shows the simulations for the five different plumes in the surroundings Mo-o. Here, a wide spread is noticeable. Similar to the random beam, the centred plume and the layered plume in the centre are measured the best. For these plumes, the plume error is reduced below 0.4. On the other end, the plume error for the double-layered plume is only reduced to 0.75. The resulting difference of 0.35 leads to a wide standard deviation. The normal double plume is measured more precisely with an error of 0.66 but is still significantly worse than for the other plumes. The single plume at the side is determined with a plume error of 0.45. This finding shows that the applied procedure has problems with disjointed plumes. The observation can also be made on the moving distances. For the two versions of the split plume, the vehicles travel the shortest distance. The observation of the distances travelled for the measurement of the other plumes show the same magnitude. The required time for the measurements is similar for all five simulation runs.

Conclusively, the best beam has the potential to reduce the plume error further than it can be done with the measuring patterns. At the same time, it provides a further optimi-

sation for the distances between the measurement locations. On the downside, it requires drastically more computation power than for the measuring patterns. This can be reduced by identifying multiple best beams before the next reconstruction loop is performed. Nevertheless, a wide spread in the results depending on the distribution of the plume in the area is noticeable. This behaviour must be analysed for its reason and minimised. Otherwise, the performance is too dependent on the expected plume in the application.

5.1.6. Comparison

At last, all discussed measuring approaches are compared. The comparison is performed with the centred plume in the Mo-o environment. The method of the measuring pattern is represented by the hourglass pattern. It shows the most promising balanced performance. The results for the plume error over the distance travelled by the UGV for the comparison are depicted in Fig. 5.19. It is noteworthy that the considered approaches outperform the random approach in regard to the travelled distance of the UGV. Due to the stagnation in the simulations with the hourglass pattern, the random measuring achieves a smaller remaining plume error than the measuring pattern. Also, the pattern method reduces the plume error at a higher rate than the best beam approach. Since the simulations are cut off at 6000 beams, it cannot be said if the best beam approach can achieve a smaller plume error than the pattern method. When the plume error is reviewed over time, the best beam approach reduces its error significantly slower than the other methods. Since prolonged

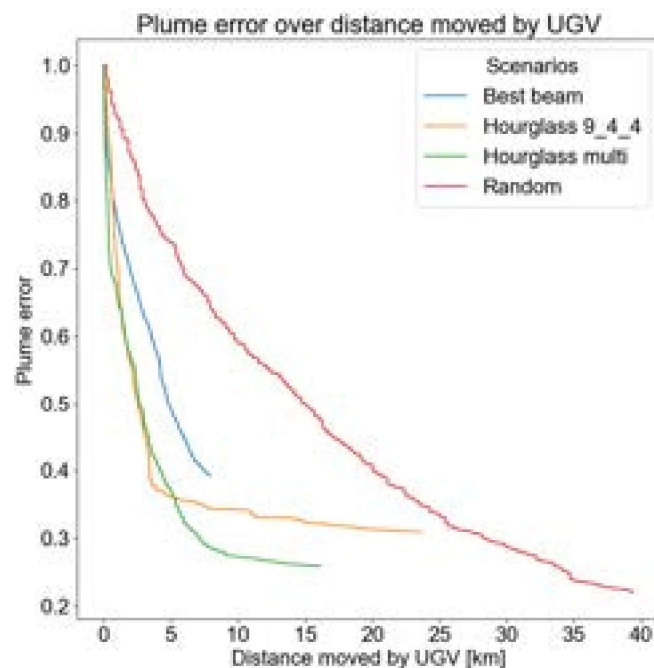
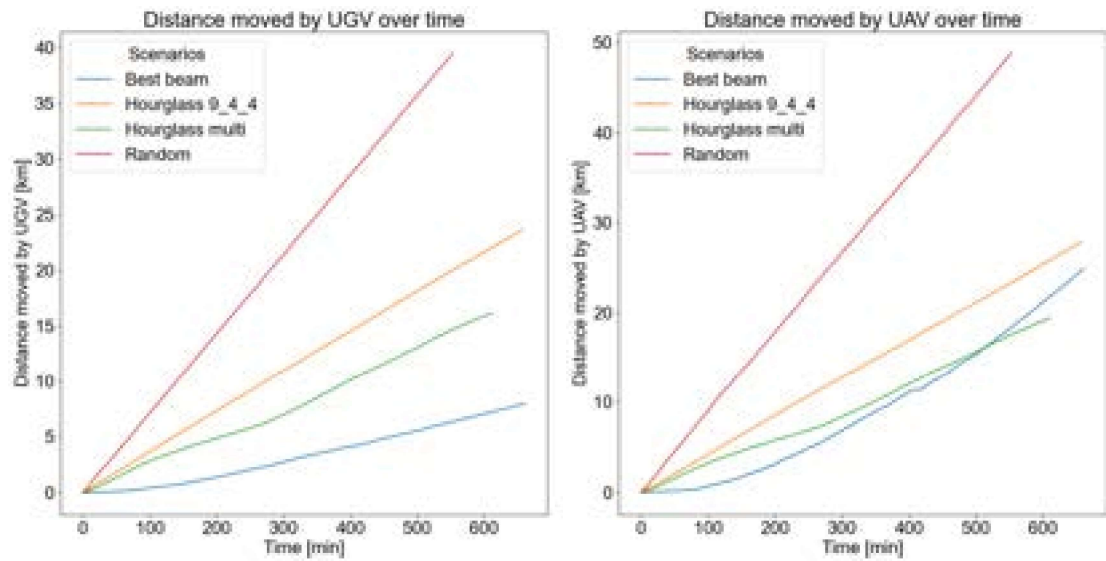


Figure 5.19.: The figure shows the different performance of the measuring methods in regard to the plume error.

measuring time is critical due to the limited battery power of the UAV, this is a strong disadvantage. However, the measuring pattern outperforms the random measurement also in regard to the time until it begins to stagnate.

At last, a short analysis on the travelling distance is necessary. In Fig. 5.20 (a) the distance for the UGV and in (b) the distance for the UAV is depicted. The plots show that the



(a) The figure shows the distances travelled by the UGV (b) The plot shows the different distances moved by the UAV for the measuring methods.

Figure 5.20.: The figure displays the difference in the travelled distance for the vehicles with the target allocation methods.

distance moved by the vehicles is distinctly lower for all concepts than for the reference of random measurements. It can also be seen that the vehicles do not have to move far during the best beam approach. This underlines the problem of the best beam approach. The required measuring time is created by a long time for the calculation of the next target. In this time period, the UGV is standing around and not consuming much battery power but the UAV has to hover all the time and slowly drains its batteries. This illustrates the need for high computation power to apply the best beam approach. The pattern approach reduces the travelled distance for both vehicles by around half in comparison to the random measurements. This is an important reduction to support the feasibility of the exploration strategy.

5.1.7. Plume Reconstruction

To conclude the section on the Python simulation, an analysis of the reconstruction of the plume is provided. For a start, the hourglass_9_4_4 pattern is investigated in the Mo-o surroundings with the centred plume. Outlined in the prior analysis, the pattern reaches its stagnation point after approximate 1000 beams. To see the differences in the recon-

struction process, the plume is reconstructed after 100, 500, and 1000 beams. To better identify voxels with small concentration values, all voxels with a concentration of 0 are marked white. Values > 0 are color coded as indicated by the legend.

In Fig. 5.21 a reconstruction of the plume after 100 measurements is shown. The Python simulation calculates 10.6 *min* to perform the measurements. The plume error is estimated with a value of 0.89 at this time. In Fig. 5.21 (a) the slice for $x = 10$ is shown. Next to it, in Fig. 5.21 (b) the section of the centre of the area for $y = 10$ is depicted. In both cases, it can be seen that only some layers have been measured at this time and low concentrations have been estimated in this area. In Fig. 5.21 (c) the horizontal layer of $z = 4$ is plotted. It already shows high concentration values in the centre. In Fig. 5.21 (d), the three-dimensional reconstruction is visualised. It shows a higher gas concentration in the centre and empty voxels measured around it. It is already possible to surmise the centred plume. In the voxels part of the beams crossing through the plume, a small

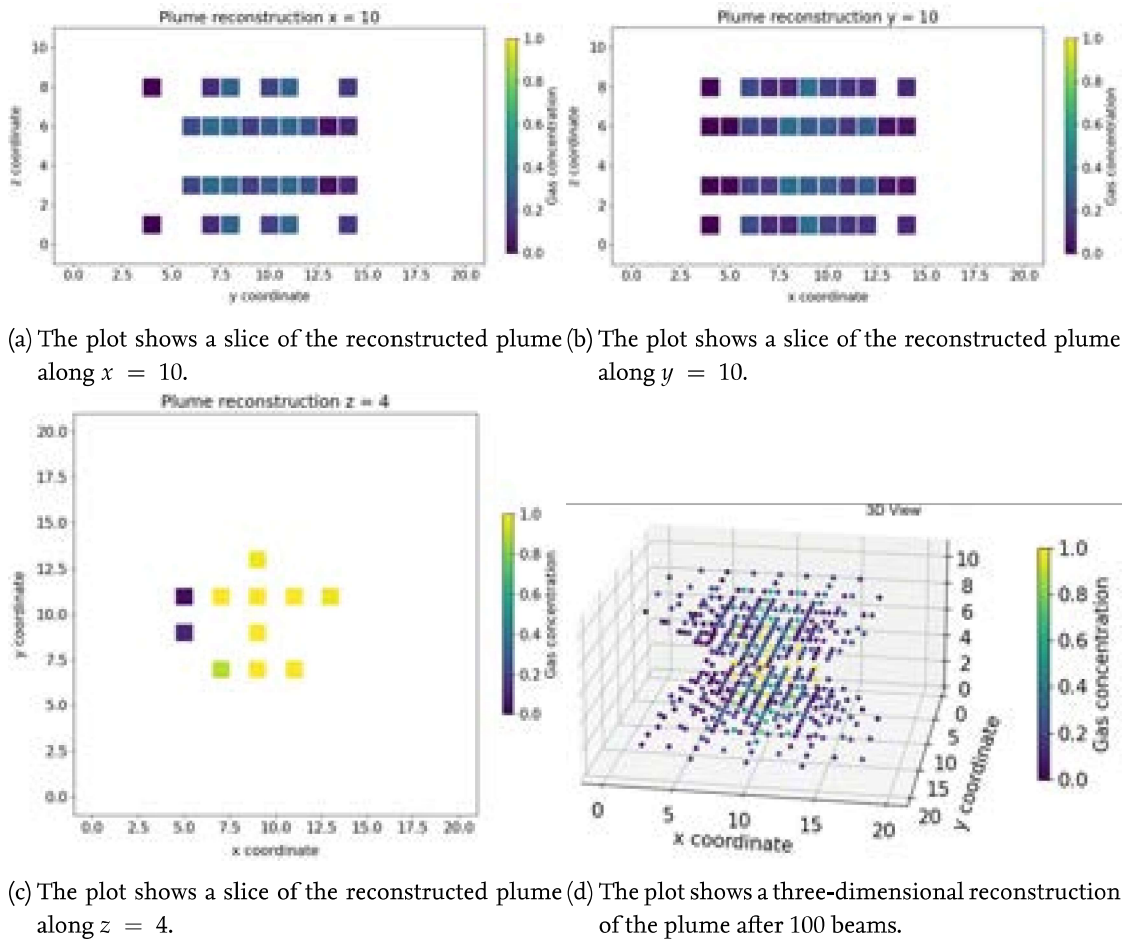


Figure 5.21.: The figure shows the plume reconstruction for a simulation with the hourglass_9_4_4 pattern in the Mo-o scenario with the centred plume after 100 beams. The plume error has a value of 0.89 at this time.

concentration is suspected and thereby the hourglass shape is formed and visible in the three-dimensional reconstruction.

In Fig. 5.22 the same simulation is depicted 400 beams later at a total of 500 measurements. At this time the plume error is 0.56 and 53.3 *min* of simulated time have passed. In the slices along $x = 10$ (Fig. 5.22 (a)) and $y = 10$ (Fig. 5.22 (b)) the centred plume is distinguishable. The concentration is estimated lower than in the simulated plume but the form of the plume differentiates clearly from the surrounding voxels. Also, along $z = 4$ (Fig. 5.22 (c)) the concentration is reconstructed in the centre and a thin border begins to be formed around it. Furthermore, the plot shows that the measurements are concentrated on the centre of the area and no gas is detected towards the border of the area. In the three-dimensional reconstruction displayed in Fig. 5.22 (d), the shape of the plume is visible and the hourglass shape of the pattern characterised the reconstruction.

In the last display of the simulation, the results of the reconstruction for 1000 beams is

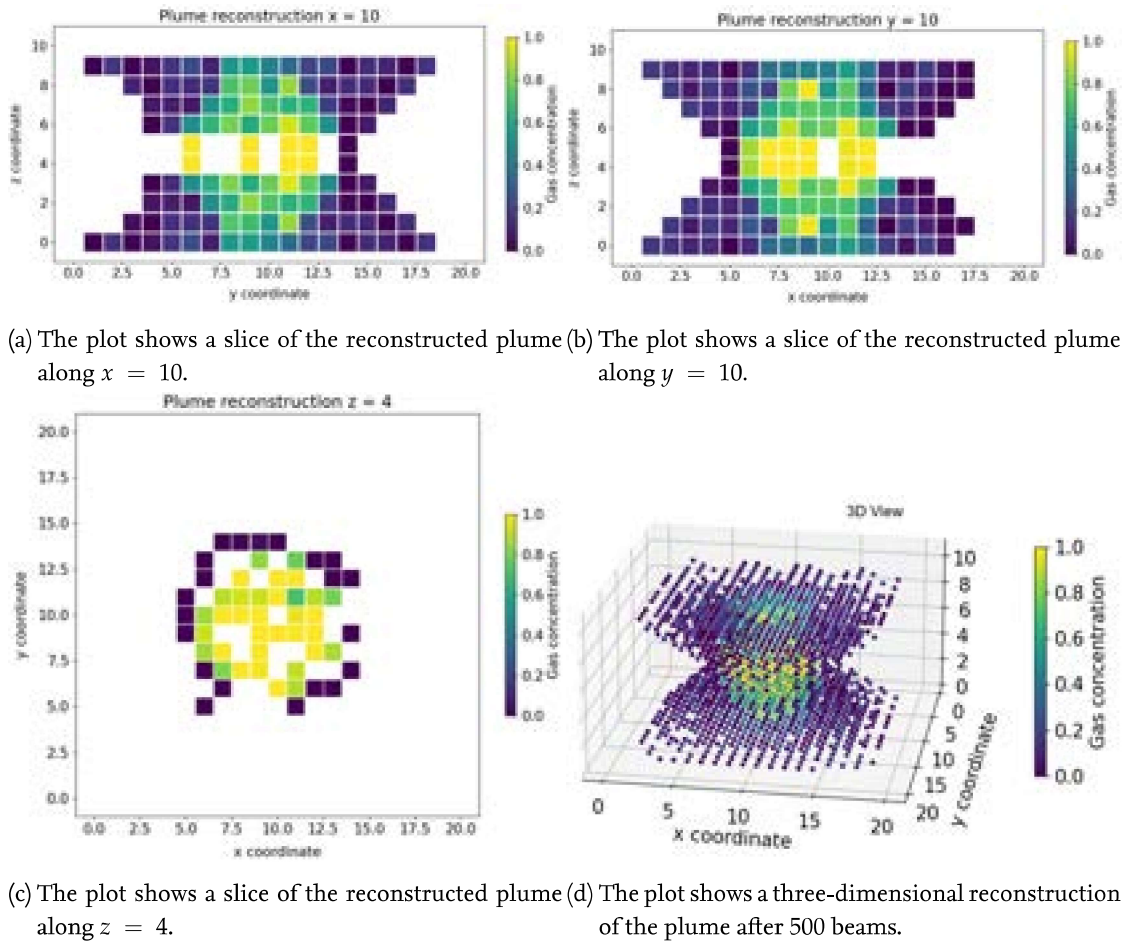


Figure 5.22.: The figure shows the plume reconstruction for a simulation with the hourglass_9_4_4 pattern in the Mo-o scenario with the centred plume after 500 measurement beams. The plume error is 0.56.

depicted in Fig. 5.23. To perform the 1000 measurements, a time of 107 *min* is estimated. At this time, the plume error is 0.37 which is close to the stagnation point for the hourglass_9_4_4 pattern. In the three slices at the position of the reconstructed plume shown in Fig. 5.23 (a), (b), and (c) the plume is visible clearly. It is noticeably distinguishable from the surrounding voxels. The same applies to the three-dimensional visualisation of the plume in Fig. 5.23 (d). Voxels that never were measured at the same time as the plume, are reconstructed with a value of 0. This shows that the reconstruction matrix does not provide sufficient information to locate the complete concentration inside the actually gas-filled voxels. Due to this lack of information it partially situates the gas in the other voxels measured in the beam. Nevertheless, the reconstruction is sufficient to be evaluated and draw a conclusion on the location of the gas.

For comparison, the reconstruction for simulations performed with the hourglass_9_4_4 pattern in the M2-2 surroundings with the layered-double plume is analysed. The results

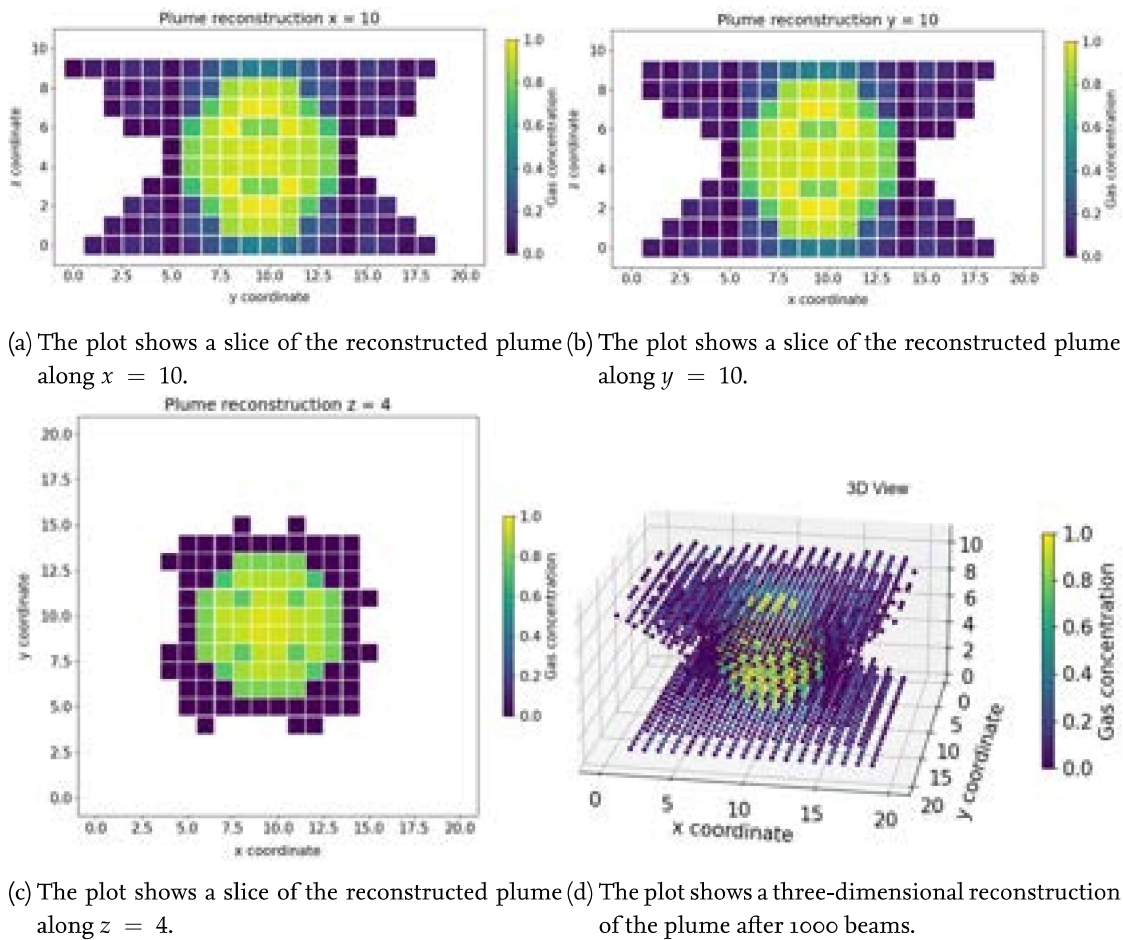


Figure 5.23.: The figure shows the plume reconstruction for a simulation with the hourglass_9_4_4 pattern in the Mo-o scenario with the centred plume after 1000 beams. The plume error at the time is 0.37.

in this scenario are poorer than in the prior reflected scenario. After 6000 performed measurements the plume error has a value of 0.517. The visualisation of the reconstructed plume shows the reason. In Fig. 5.24 the reconstruction after 100 beams is depicted. It shows that the first plume has been discovered. At this time the plume error has a value of 0.98 and the vehicles required 10.6 *min* to perform the 100 measurements.

In Fig. 5.25 the results of the reconstruction after 500 beams are shown. In all four visualisations in the figure, it can be seen that the measurements are focused around the discovered plume. With the 500 performed measurements, the vehicles obtained a plume error of 0.867 after 53.4 *min*. The size of the pattern makes the measurements towards the border of the area unattractive because not the full set of valid beams can be created. The result is visible in the slices of the area shown in Fig. 5.25 (a), (b), and (c). It is visible that the knowledge about the voxels of the plume towards the border of the area is poorer than towards the centre. However, the form and the layers of the first plume become clear. This

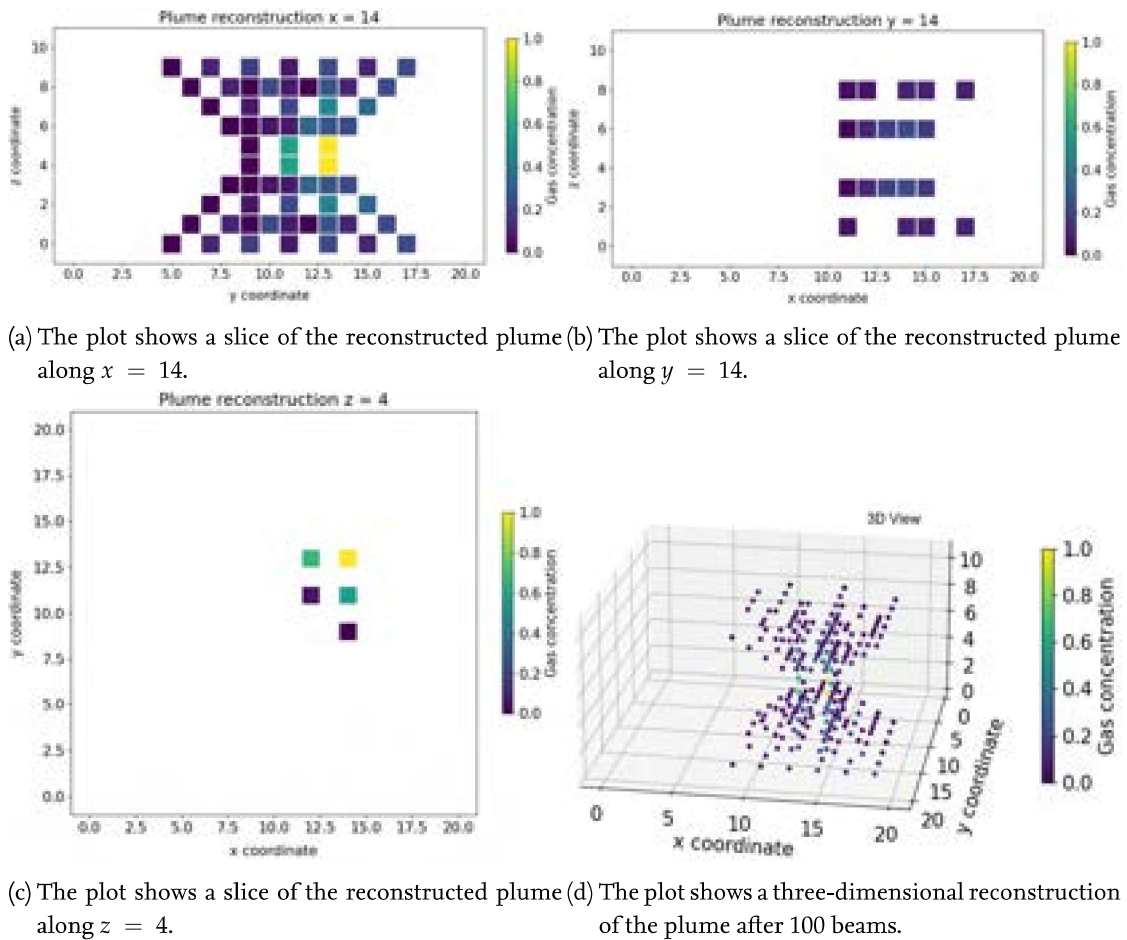


Figure 5.24.: The figure shows the plume reconstruction for a simulation with the hourglass_9_4_4 pattern in the M2-2 scenario with the layered double plume after 100 beams. The plume error has a value of 0.98 at this time.

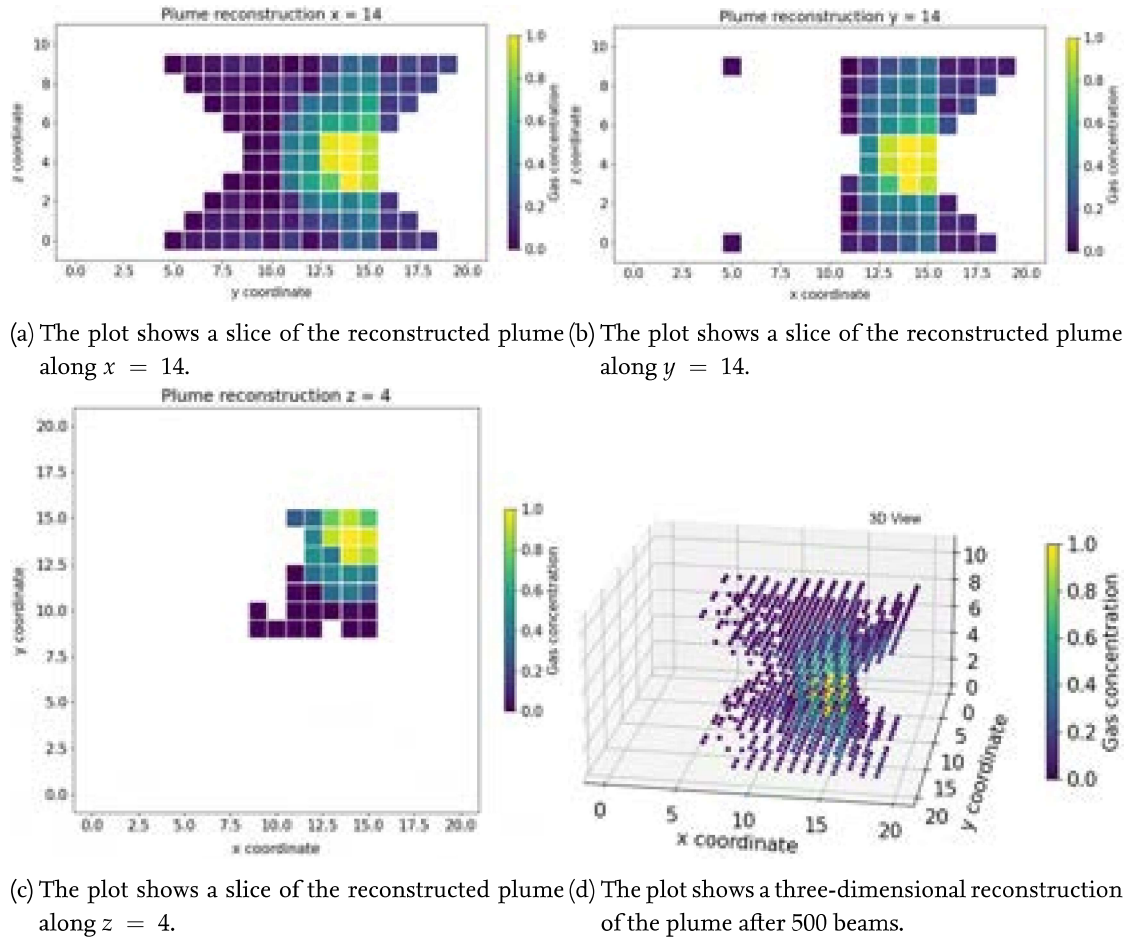


Figure 5.25.: The figure shows the plume reconstruction for a simulation with the hourglass_9_4_4 pattern in the M2-2 scenario with the layered double plume after 500 beams. The plume error is 0.867.

is the case for the slices as well as for the three-dimensional representation. Nevertheless, until this point, the system has no knowledge about the second plume.

Next, Fig. 5.26 shows the reconstruction of the plume after 1000 measurements. As a reminder, at that time the centred plume was known with an error of 0.37. The measurement of the double layered plume attains only a plume error of 0.707 after the 1000 measurements. To perform these measurements, the system takes 106.75 *min*. The time is similar to the centred plume. However, the depiction of the reconstructed plume in Fig. 5.26 shows the reason for the high plume error. The system found the second plume and measured it to the same grade as the first plume. Still, the knowledge about the voxels towards the outer side of the area is low. But the layers of the plume get visible. Although there are two plumes, the concept managed to detect both in the same amount of time as the single source. But the degree of reconstruction is not as good as for the single plume. However, this is also related to the proximity of the border. With an enlarged area, the

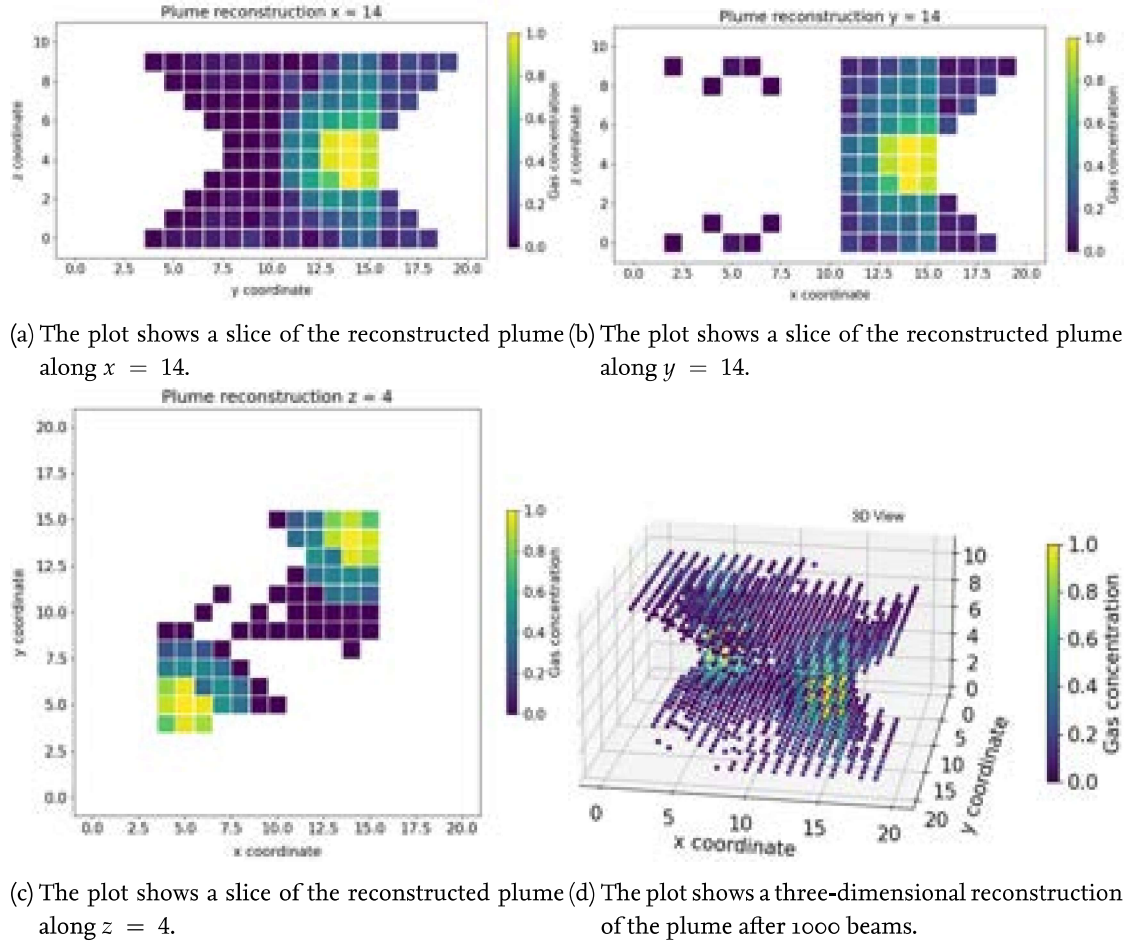


Figure 5.26.: The figure shows the plume reconstruction for a simulation with the hourglass_9_4_4 pattern in the M2-2 scenario with the layered double plume after 1000 beams. The plume error at the time is 0.707.

measurement of the double plume has to be re-evaluated. It has to be checked if the strategy manages to measure both plumes in sufficient time if it is allowed to measure them with the full patterns.

Until the end of the simulation, 6000 measurements are performed. The final reconstruction is shown in Fig. 5.27. With the 6000 measurements the plume error is 0.517 after 666.9 min. This is a long time, especially in regard to the limited flight time of the UAV. In the plot of the x-slice and in the z-slice the obstacle next to the upper right plume is clearly distinguishable. It also shows that the strategy allowed measuring the plume even though it is located directly next to the obstacle. Furthermore, the result for the upper left plume is similar to the knowledge about the second plume which is not close to an obstacle. However, the measurements caused a blurring in the estimation of the layers of the plumes. The concentration in the centre is reduced compared to the prior displayed reconstruction.

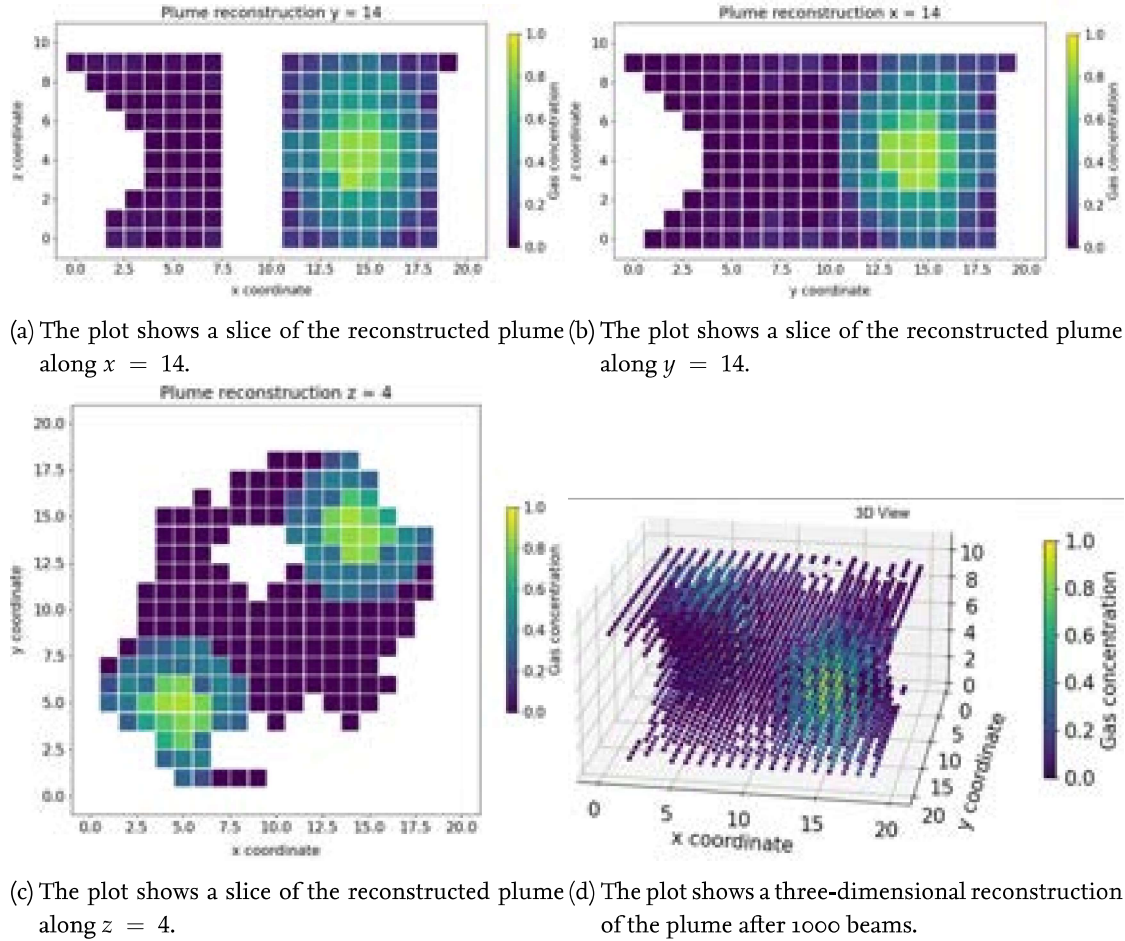


Figure 5.27: The figure shows the plume reconstruction for a simulation with the hourglass_9_4_4 pattern in the M2-2 scenario with the layered double plume after 6000 beams. The plume error at the time is 0.517.

Overall, it can be taken along from the Python simulations that the concept, developed in this study, for the exploration strategy is applicable. The engineering approach with the heuristic is verified. A set of parameters is identified and different measuring methods are examined. It is possible to reconstruct plumes in different scenarios but the quality of the reconstruction depends on the location of the plume in the area. With these results, it is possible to transfer the concept to the physics simulation and test the implementability.

5.2. Physics Simulation Results

In this section, the findings of simulations performed with Gazebo and ROS are presented. With the simulation, it was verified that the modified software from the Python simulation is compatible with the actual hardware. In Fig. 5.28 a screenshot of the physics simulation is displayed. On the right side, the Gazebo simulator is displayed. It shows the

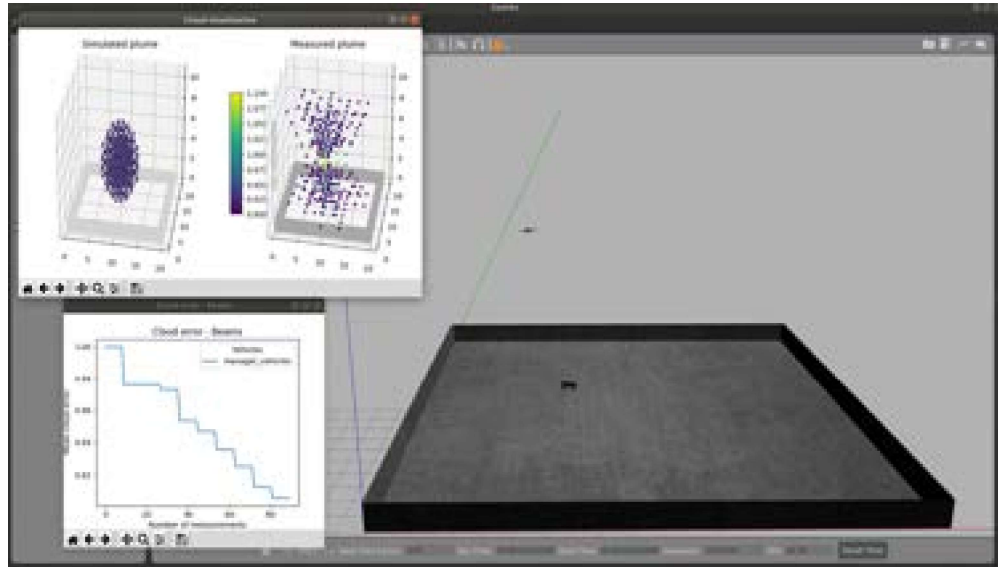


Figure 5.28.: The figure shows a screenshot of the physics simulation. On the right, the vehicles moving in the Gazebo simulation are displayed. In the lower-left corner, the current error of the reconstructed plume is presented. On the upper left side, the current knowledge about the plume is compared to the actual plume.

UGV and the UAV performing measurements in the Mo-o surroundings. In the upper left corner, the simulated plume (left) is compared to the currently reconstructed plume (right). The plumes are visualised with the matplotlib library. The visualisation of the simulated plume shows that it is the single-layered centred plume. In the lower-left corner, the current plume error is plotted over the performed measurements on the abscissas.

A finding during the simulations was the underestimation of the movement in the Python simulation. The movement of the vehicles requires more time than expected. A reason for that is the A* algorithm. The UAV is sent from voxel to voxel by the A* algorithm. Since the UAV tries to reach the commanded point and not overshoot, it does not accelerate much. Thus, the UAV does not take advantage of its moving speed. The result is that it arrives at the target after a similar time as the UGV and not substantially earlier as expected. With a more complex path planner, the problem can be solved and the expected behaviour can be achieved. Due to the absence of such complex planner, the A* algorithm is also used for the proof-of-concept demonstration for the UAV. The same problem is observed with the UGV. Especially for narrow turns, the UGV has difficulties moving to the target position. To improve the movement, the pose commanded to the UGV includes the orientation to the next commanded position known. Since this does not result in a satisfying improvement, the path planner provided by Robotnik is used for the commanding of the UGV. Therefore, only the target position is sent to the UGV. Further, a bitmap containing the surroundings is submitted to the obstacle detection of the UGV. This results in a different problem. The path planner of the UGV includes a

safety margin around obstacles. If it is chosen too big, the UGV is not able to move into voxels next to the obstacles and cannot measure from this position. The path planner of the UGV calculates the distances to obstacles in relation to its centre of mass. If the safety margin is chosen too small, the back or the front of the UGV can collide with the obstacles. Due to the nature of the obstacle generation, they are formed like ramps. It can occur that the UGV gets to an upright position and thereby cannot move any more.

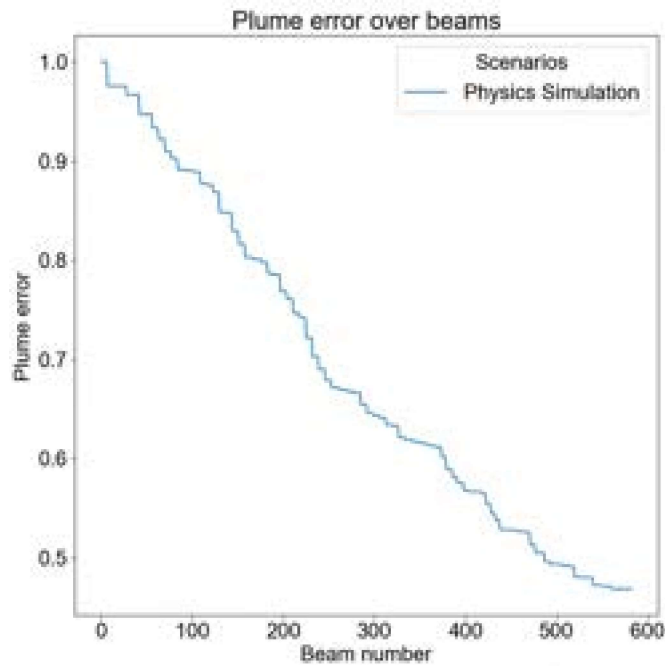


Figure 5.29.: The figure shows the trend of the plume error during the physics simulation.

In addition, it is shown that the simulated TDLAS sensor is functional. The analysis of a recording of a run of the physics simulation performed with the `hourglass_9_4_4` pattern in the Mo-o surroundings with the centred plume confirms the misjudgement for the required time to perform the measurements. During the simulation, 582 measurements are performed resulting in a plume error of 0.47. The course of the plume error is depicted in Fig. 5.29. It is as expected from the Python simulation. However, with the Python simulation, 62 *min* are estimated to perform the 582 measurements. In the physics simulation, 127.85 *min* are required to perform the measurements. This is more than double the time estimated. Hence it is important to adjust the assessment for the required time in the Python simulation for further analysis. This is essential because it correlates with the limited flight time of the UAV. The estimated measurement time has an important impact on the number of measurements that can be obtained from a single UAV battery charge.

Conclusively, it can be established that the exploration strategy can be realised with the envisaged vehicles. For better results, it is necessary to improve the capabilities of path planning for the vehicles. Also, it is necessary to adjust the assessment of required time

in the Python simulation.

5.3. Field Test Realisation and Results

In the field tests, the implementability of the concept is shown. The vehicles receive positions from the planning node and perform cooperative measurements as planned by the algorithm. In Fig. 5.30 an image of the demonstration execution is displayed. In the centre of the image, the UAV can be observed during the movement. Towards the left, the UGV moves to the next measurement position. In the lower right corner, the UGV base station is located.

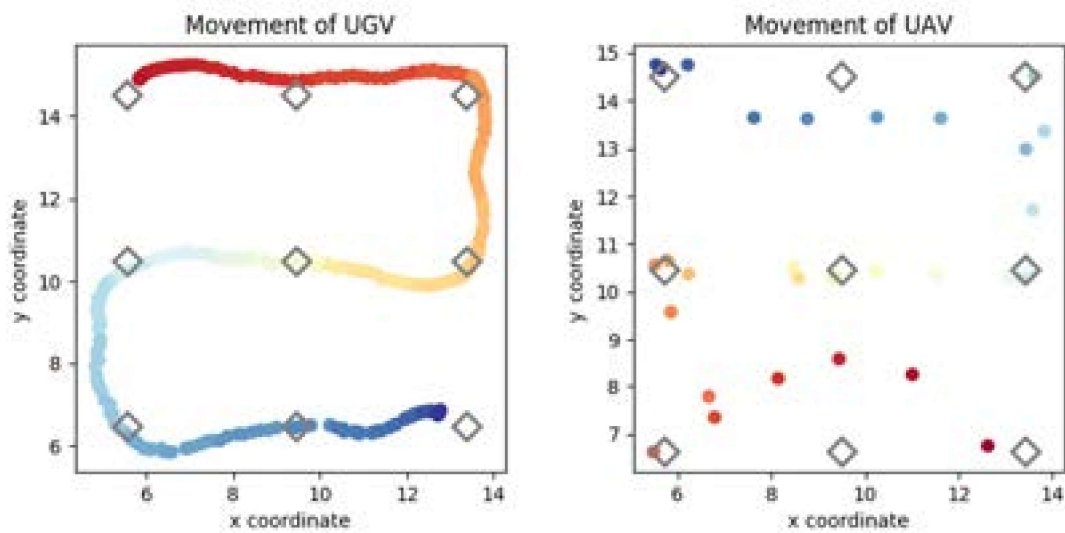
Further, multiple observations are made which confirm the findings of the physics simulations. It is important to take the turn radius of the UGV into account because it significantly influences the movement speed. If the turn radius does not allow direct movement to the next commanded measuring point, the UGV performs a complex turn manoeuvre. In addition, the usage of the A*-algorithm for the path planning of the UAV results in a rather slow movement of the UAV because it receives a waypoint list with a distance of approximately 1 m between the waypoints. Since the UAV tries to arrive at the waypoints with a forward velocity of 0 m/s, it does not accelerate to high movement speeds. As a result, the movement times for the UGV and the UAV are similar. This fact shows, that a high-quality path planner which takes the movement capabilities of the vehicles into account, influences the results significantly. An improved path planner leads to shorter



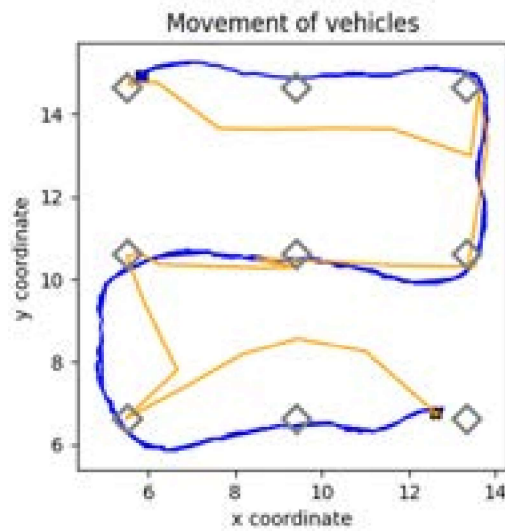
Figure 5.30.: The figure displays an image taken during the execution of the field tests. On the lower right, the RTK system of the UGV is located. In the centre of the image, the UAV flies in the exploration area. On the left, the UGV moves inside the area.

intervals between the measurements. As shown in the simulations, a large number of measurements have to be performed. If the time between the measurements can be reduced, a crucial improvement can be achieved. The more measurements can be performed in a short amount of time, the fewer changes due to environmental influences like wind or man-made influences, impact the measuring result. Man-made influences can be people or vehicles moving through the area and introducing turbulences into the plume.

In Fig. 5.31 a section of the paths of the vehicles is displayed and overlapped with the measurement positions of the pattern in the form of diamonds. The paths are recorded during the performance of the hourglass pattern 9_4_4. In (a) the path of the UGV is illustrated. It begins at the red-marked position in the upper left and then follows the pattern in an inverted s-shape until it reaches the blue-marked lower right corner. The path of the UAV is displayed in (b). In the plot, it can be seen that the position of the UAV is published in a lower rate than the position of the UGV. Therefore, the data points are further apart. The UAV starts in the lower right corner and follows the inverted s-shape to the upper left corner. The colors in (a) and (b) show the chronological sequence. From this, it can be seen, that the vehicles are always on the opposite side of the pattern. The centre of the pattern is an exception. In Fig. 5.31 (c), the paths of the two vehicles overlap. It can be seen that the UGV determines its position with higher precision than the UAV. The reason is the RTK system. This finding demonstrates the relevance of an accurate positioning system. To gain a high-resolution map of the gas distribution in the area, it is necessary to be able to determine the position of the vehicles with high precision. In the present example, the pursued resolution of the gas map is in the magnitude of 1 m. To achieve this, positioning accuracy of less than 1 m is necessary. The accuracy of GPS alone allows a precision of 2 m. With the RTK-system, it can be improved to the magnitude of millimetres. Therefore, either the envisaged resolution of the gas plumes needs to adapt or a GPS solution alone is not sufficient.



(a) Section of the path of the UGV. The UGV starts at the red marked location and ends at the dark blue location.
 (b) Section of the path of the UAV. The UAV starts at the red marked location and ends at the dark blue location.



(c) Overlay of the paths of the vehicles. The UGV is displayed in blue and the UAV in orange.

Figure 5.31.: The figure shows a presentation of a section of the vehicles path during the proof-of-concept demonstration. The diamonds mark the measurement positions.

6. Summary

At the beginning of the present thesis an introduction is given to the relevant theoretical background the thesis is based on. This includes a classification in the frame of the state-of-the-art in the field of unmanned exploration. Further, the measuring method TDLAS is introduced as the thesis is based on this technique. In addition, the A* algorithm, the travelling salesperson problem, and the Bresenham line algorithm are presented. These are existing algorithms used in the scope of the present thesis.

Afterwards, the new exploration strategy developed in the scope of this thesis is presented. At first, a detailed analysis of the problem is performed. It is pointed out that the new approach combining unmanned ground and airborne vehicles allows the exploration of a three-dimensional area which is not possible with a solely ground-based approach. Furthermore, the TDLAS method allows the measurement of gas concentrations from afar without directly interacting with the plume. This enables the measurement without introducing distortion during the process. In the developed concept, the UGV is equipped with the sensor and detector for the measurement. To reflect the beam back to the UGV, the UAV is carrying a retro-reflective surface. Out of the TDLAS measurements, the plume is reconstructed with a linear least square optimisation approach. The TDLAS method requires measurements from a multitude of different angles but the battery power of the vehicles is limited. Therefore, in the scope of this thesis, an algorithm is developed to identify informative measuring targets. The algorithm is based on an engineered heuristic to identify the measurement location. The heuristic contains an exploitation term considering the current gas concentration, an exploitation term considering previous measurements, and an addend taking the costs for movement into account. They are weighted, so one can adjust the behaviour.

For the verification of the exploration strategy, a multi-step approach is performed. In the first step, the concept is implemented in a Python-based simulation. With the simulation, the parameters used in the heuristic are identified and the basic operating principle of the concept is verified. Afterwards, different methods for the arrangement are performed and evaluated. The outcome of these analyses is that single measurements require high computing power. If this computing power is not available, it leads to slow information gathering. This is suboptimal because the vehicles, especially the UAV, still consume battery power. If the battery of the UAV has to be changed, the process is prolonged further. For this reason, it is recommended to perform the measurements in pre-defined patterns. A series of different patterns are investigated with the Python simulation. The most promising pattern identified is the so-called hourglass pattern. In this pattern, the vehicles are positioned at the opposite side of a rectangle and perform measurements through the centre. With this pattern, the plume error after the reconstruction based on the mea-

surements is reduced significantly while requiring an acceptable travelling distance for the vehicles. Furthermore the simulations show the importance of using a multitude of different measuring angles to improve the reconstruction result. However it is important to find a balance between the number of measuring angles and the increase of calculation time due to fact that more possibilities are taken into account. In total more than 530 simulation runs are executed to identify the preferred setup for the measurement of the plume.

In the next step, the functionality of the Python simulation is transferred to a physics simulation based on ROS and Gazebo. It utilises software in the loop simulation of the UGV and the UAV. The simulation is used to take the constraints of the actual hardware into account. It is used to test the interaction of the exploration strategy, in detail the measurement location determination algorithm, with the actual vehicles. The physics simulation shows that the performance of the vehicles is overestimated in the Python-based simulation. The vehicles require much more time to move to the target positions than anticipated. Conclusively it is established that improved path planners for the vehicles are required to perform more advanced simulations of the exploration strategy. Nevertheless, the compatibility of the implementation of the exploration strategy with the vehicles was verified.

In the third and last step, a proof-of-concept demonstration with the actual available vehicle hardware is performed. The sensor and the gas measurements are simulated in software-in-the-loop simulations. The demonstration verifies the realisation with the actual hardware (i.e. UAV and UGV). However, the demonstration substantiates the known issues from the physics simulation. In addition, it exhibits the need for precise positioning of the vehicles. Accurate positioning is necessary to ensure the cooperation capabilities of the vehicles. Further, it is apparent that the need for a multitude of measurements is in conflict with the limited flight time of the UAV.

7. Conclusion

The thesis sets out the developed exploration strategy, verifies its feasibility and points out its shortcomings. Concluding recommendations and necessary steps for further work in the exploration of gas plumes with a cooperation of ground-based and airborne vehicles by remote sensing is provided.

The demonstration shows that the exploration strategy is applicable and implemented. For a next step, it is required to acquire hardware for the alignment of the sensor. This includes hardware required to precisely identify the position of the reflective surface at the UAV as well as a gimbal to perform the pointing of the sensor. Then the implementation of the exploration strategy has to be retested with the new hardware. To support these tests, the Gazebo and ROS simulations can be expanded by simulations of the gimbal, the sensor for the tracking of the UAV, and a model of the actual TDLAS sensor. With the simulation, the functionality can be verified and the performance improved before it is transferred to the actual hardware. This reduces the risk of damaging the hardware.

Once all systems required for the concept are implemented and tested, it is recommended to expand the measuring principle to a swarm of vehicles. In a first step, more UAVs with retro-reflectors can be added. The prerequisite for this approach is that the tracking and pointing system of the sensor allow a fast and precise change to another target.

A different aspect for investigation is performing measurements during the movement of the vehicles. This requires a high precise pointing system able to compensate for the motion of the UGV during the movement. In addition, the tracking system must be able to accurately pursue the UAV during its movement. Furthermore, it must be able to quickly re-establish a target lock after losing the visual line of sight between the vehicles. Introducing more UAVs into the system, decreases the time required for the plume error reduction. Additionally, small UAVs are an inexpensive extension of the concept.

For the experiments with the actual hardware it should be considered to enlarge the size of the voxels. As it has been stated, the positioning of the vehicles is important. With an accuracy of 2 m at best for the GPS, it is not reasonable to set the voxel size to 1 m. This is only possible while using an RTK system. In return, the area can be enlarged without increasing the computing effort.

Beforehand it is suggested to further improve the measuring position acquisition. As proposed in the result section, it should be examined to include the measuring angle into the cost function. To perform more advanced tests of the concept in the physics simulation, it will be necessary to improve the path planners used for the vehicles. Another factor to improve the measurement positions is the consideration of the propagation behaviour of gases. With this knowledge, the exploitation part of the heuristic is improved.

Bibliography

- [1] Bulat Abbyasov et al. "Automatic tool for Gazebo world construction: from a grayscale image to a 3D solid model". In: 2020 *IEEE International Conference on Robotics and Automation (ICRA)*. 2020, pp. 7226–7232. DOI: 10.1109/ICRA40945.2020.9196621.
- [2] *Advantages and Disadvantages of Remote Sensing*. website. Last retrieved: 09.07.2022. URL: <https://grindgis.com/remote-sensing/advantages-and-disadvantages-of-remote-sensing>.
- [3] United States Environmental protection agency. *Basic Information about Landfill Gas*. website. Last retrieved: 08.07.2022. URL: <https://www.epa.gov/lmop/basic-information-about-landfill-gas>.
- [4] Tiffany Wang Alex Chumbley Karleigh Moore. *Traveling Salesperson Problem*. website. Last retrieved: 09.07.2022. URL: <https://brilliant.org/wiki/traveling-salesperson-problem/>.
- [5] John Amanatides and Andrew Woo. "A Fast Voxel Traversal Algorithm for Ray Tracing". In: *Proceedings of EuroGraphics 87* (Aug. 1987).
- [6] Muhammad Asif Arain et al. "The right direction to smell: Efficient sensor planning strategies for robot assisted gas tomography". In: 2016 *IEEE International Conference on Robotics and Automation (ICRA)*. 2016, pp. 4275–4281. DOI: 10.1109/ICRA.2016.7487624.
- [7] PX4 Autopilot. *Simulation*. Website. Last retrieved: 16.07.2022. URL: <https://docs.px4.io/v1.12/en/simulation/>.
- [8] PX4 Autopilot. *Gazebo Simulation*. Website. Last retrieved: 16.07.2022. URL: <https://docs.px4.io/v1.12/en/simulation/gazebo.html>.
- [9] J. E. Bresenham. "Algorithm for computer control of a digital plotter". In: *IBM Systems Journal* 4.1 (1965), pp. 25–30. DOI: 10.1147/sj.41.0025.
- [10] The SciPy community. *scipy.optimize.nnls*. Website. Last retrieved: 16.07.2022. URL: <https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.nnls.html>.
- [11] The SciPy community. *scipy.sparse.linalg.lsqr*. Website. Last retrieved: 16.07.2022. URL: <https://docs.scipy.org/doc/scipy/reference/generated/scipy.sparse.linalg.lsqr.html>.
- [12] Thomas H. Cormen et al. *Algorithmen - Eine Einführung*. De Gruyter Oldenbourg, 2017. ISBN: 9783110522013. DOI: 10.1515/9783110522013. URL: <https://doi.org/10.1515/9783110522013>.

-
- [13] Cubic. *TDLAS CH₄ Sensor*. website. Last retrieved: 12.04.2022. URL: https://en.gassensor.com.cn/FlammableGasSafetyMonitoringSensor/info_itemid_1728.html.
- [14] Cubic-Ruiyi. *GasTDL-3100*. website. Last retrieved: 12.04.2022. URL: https://www.gas-analyzers.com/ProcessGasAnalyzer/info52/?gclid=Cj0KCQjwxtSSBhDYARIsAEn0thRYzHlz0wh1Awepblnb6bS8vN0lVDG76GvPH_acGga_ep3zaSvT-M0aAvW8EALw_wcB.
- [15] DJI Enterprise. *U10 UAV BASED LASER METHANE LEAKAGE DETECTOR*. website. Last retrieved: 12.04.2022. URL: https://terra-1-g.djicdn.com/851d20f7b9f64838a34cd02351370894/U10_Gas_Detector_EN.pdf.
- [16] Exprt environmental. *FPI - Model RLGD-100 - Remote Laser Gas Detectors*. website. Last retrieved: 12.04.2022. URL: <https://www.environmental-expert.com/products/fpi-model-rlgd-100-remote-laser-gas-detectors-458663>.
- [17] Open Source Robotics Foundation. *Gazebo - Robot simulation made easy*. website. Last retrieved: 08.07.2022. URL: <http://gazebosim.org/>.
- [18] Qiang Gao et al. "Tunable multi-mode diode laser absorption spectroscopy for methane detection". In: *Sensors and Actuators A: Physical* 199 (2013), pp. 106–110. ISSN: 0924-4247. DOI: <https://doi.org/10.1016/j.sna.2013.05.012>. URL: <https://www.sciencedirect.com/science/article/pii/S0924424713002586>.
- [19] Gremsy. *Gremsy T3V3 Spec*. website. Last retrieved: 13.04.2022. URL: <https://gremsy.com/gremsy-t3v3>.
- [20] Genevieve Hayes. *Solving Travelling Salesperson Problems with Python*. website. Last retrieved: 09.07.2022. URL: <https://towardsdatascience.com/solving-travelling-salesperson-problems-with-python-5de7e883d847>.
- [21] Haijie He et al. "In-Situ Testing of Methane Emissions from Landfills Using Laser Absorption Spectroscopy". In: *Applied Sciences* 11.5 (2021). ISSN: 2076-3417. DOI: 10.3390/app11052117. URL: <https://www.mdpi.com/2076-3417/11/5/2117>.
- [22] Elizabeth Howell and Vicky Stein. *Mars missions: A brief history*. website. Last retrieved: 08.07.2022. URL: <https://www.space.com/13558-historic-mars-missions.html>.
- [23] Infrared cameras Inc. *Innovative gas detection solution*. website. Last retrieved: 12.04.2022. URL: <https://f.hubspotusercontent10.net/hubfs/20335613/ici-tdlas-data-specifications.pdf>.
- [24] Infrared cameras Inc. *Innovative gas detection solution*. website. Last retrieved: 12.04.2022. URL: <https://infraredcameras.com/products/tunable-diode-laser-spectroscopy-tdlas>.

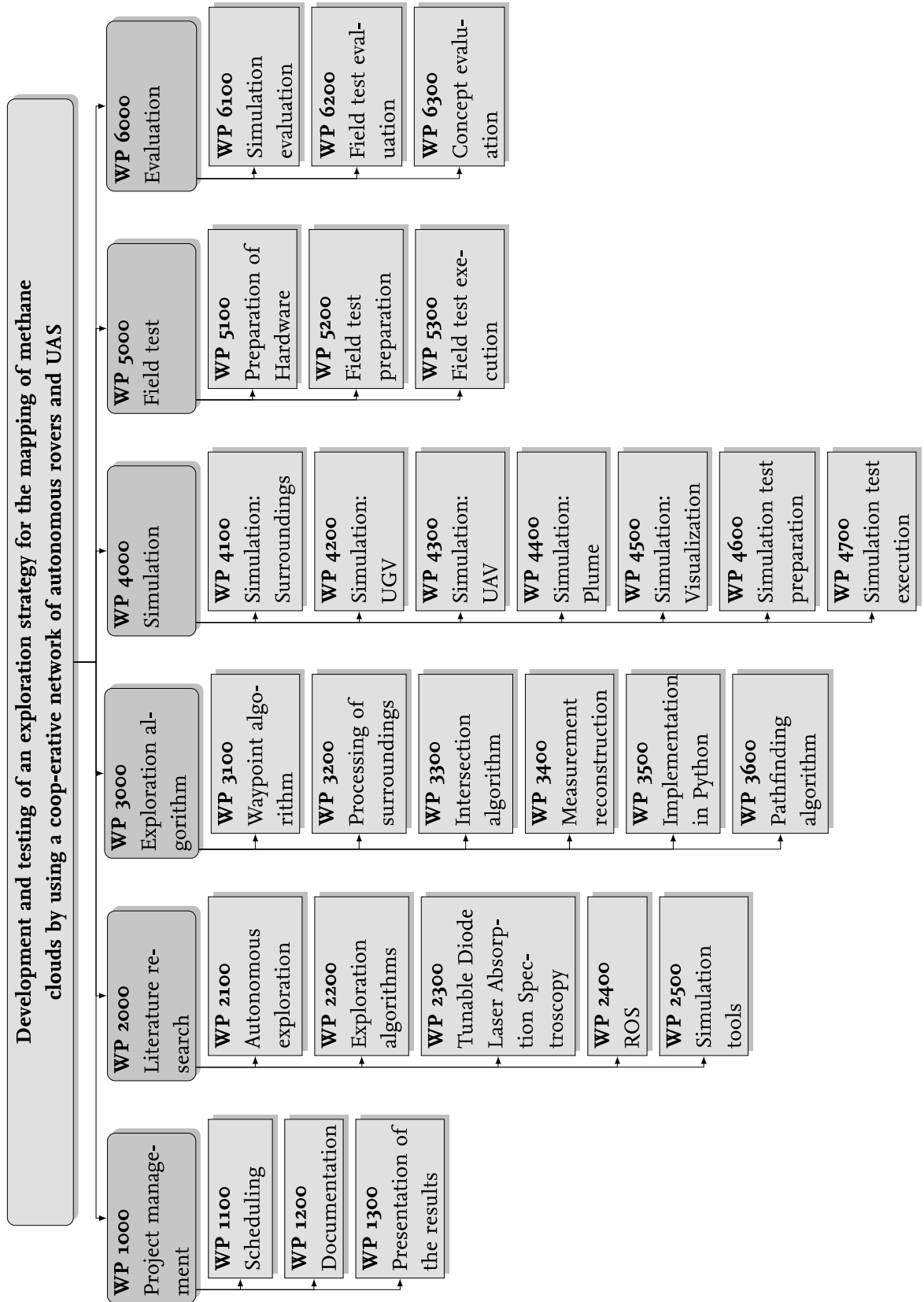
- [25] Andrew Jones. *Applying the A* Path Finding Algorithm in Python (Part 1: 2D square grid)*. Website. Last retrieved: 16.07.2022. Sept. 2018. URL: <https://www.analytics-link.com/post/2018/09/14/applying-the-a-path-finding-algorithm-in-python-part-1-2d-square-grid>.
- [26] Linus Luotsinen, Avelino Gonzalez, and Ladislau Bölöni. "Collaborative UAV Exploration of Hostile Environments". In: (Dec. 2004), p. 7.
- [27] Andreas Meister. *Numerik linearer Gleichungssysteme*. 1st ed. Springer Fachmedien Wiesbaden: Vieweg+Teubner Verlag Wiesbaden, 1999. ISBN: 978-3-322-93899-2. DOI: <https://doi.org/10.1007/978-3-322-93899-2>.
- [28] Andrew Myers. *Stanford-led study: Methane leaks are far worse than estimates, at least in New Mexico, but there's hope*. website. Last retrieved: 08.07.2022. URL: <https://news.stanford.edu/2022/03/24/methane-leaks-much-worse-estimates-fix-available/>.
- [29] nanoplus. *Tunable Diode Laser Absorption Spectroscopy (TDLAS)*. website. Last retrieved: 09.07.2022. URL: <https://nanoplus.com/en/technology/tdlas/>.
- [30] NASA. *Mars 2020 Mission Perseverance Rover - Communications*. website. Last retrieved: 08.07.2022. URL: <https://mars.nasa.gov/mars2020/spacecraft/rover/communications/>.
- [31] NASA. *Mars Helicopter*. website. Last retrieved: 08.07.2022. URL: <https://mars.nasa.gov/technology/helicopter/#>.
- [32] NASA. *Voyager - Mission Status*. website. Last retrieved: 08.07.2022. URL: <https://voyager.jpl.nasa.gov/mission/status/>.
- [33] NASA. *What is Remote Sensing?* website. Last retrieved: 09.07.2022. URL: <https://www.earthdata.nasa.gov/learn/backgrounders/remote-sensing>.
- [34] NASA and Brian Dunbar. *First Human-Made Object to Enter Space*. website. Last retrieved: 08.07.2022. URL: https://www.nasa.gov/mission_pages/explorer/bumper.html.
- [35] Patrick P. Neumann et al. "Aerial-based gas tomography – from single beams to complex gas distributions". In: *European Journal of Remote Sensing* 52.sup3 (2019), pp. 2–16. DOI: 10.1080/22797254.2019.1640078. eprint: <https://doi.org/10.1080/22797254.2019.1640078>. URL: <https://doi.org/10.1080/22797254.2019.1640078>.
- [36] Patrick P. Neumann et al. "Gas Tomography Up in the Air!" In: 2018 *IEEE SENSORS*. 2018, pp. 1–4. DOI: 10.1109/ICSENS.2018.8630293.
- [37] NOAA. *What is Remote Sensing?* website. Last retrieved: 09.07.2022. URL: <https://oceanservice.noaa.gov/facts/remotesensing.html>.

- [38] Knöbl-Knor OG. *DJI Enterprise U10 Laser Methan Gas Detektor für DJI Matrice M300 / M200 Serie PSDK*. website. Last retrieved: 12.04.2022. URL: https://www.meine-drohne.at/index.php?id_product=200&rewrite=dji-mavic-mini-fly-more-combo-drohne-mit-12mp-27k-kamera-quadrokoetter-249g&controller=product.
- [39] Amit Patel. *Heuristics*. website. Last retrieved: 09.07.2022. URL: <http://theory.stanford.edu/~amitp/GameProgramming/Heuristics.html>.
- [40] Amit Patel. *Introduction to A**. website. Last retrieved: 18.04.2022. URL: <http://theory.stanford.edu/~amitp/GameProgramming/AStarComparison.html>.
- [41] Chao-Chung Peng and Chao-Yung Hsu. "Integration of an unmanned vehicle and its application to real-time gas detection and monitoring". In: *2015 IEEE International Conference on Consumer Electronics - Taiwan*. 2015, pp. 320–321. DOI: 10.1109/ICCE-TW.2015.7216921.
- [42] Anna Rabajczyk et al. "Monitoring of Selected CBRN Threats in the Air in Industrial Areas with the Use of Unmanned Aerial Vehicles". In: *Atmosphere* 11 (Dec. 2020), p. 1373. DOI: 10.3390/atmos11121373.
- [43] Open Robotic. *ROS Concepts*. Website. Last retrieved: 15.06.2022. June 2014. URL: <http://wiki.ros.org/ROS/Concepts>.
- [44] Open Robotic. *ROS Introduction*. Website. Last retrieved: 15.06.2022. Aug. 2018. URL: <http://wiki.ros.org/ROS/Introduction>.
- [45] Open Robotics. *About ROS*. website. Last retrieved: 08.07.2022. URL: <http://wiki.ros.org/de/ROS/Introduction>.
- [46] Maurizio Rossi and Davide Brunelli. "Gas Sensing on Unmanned Vehicles: Challenges and Opportunities". In: *2017 New Generation of CAS (NGCAS)*. 2017, pp. 117–120. DOI: 10.1109/NGCAS.2017.58.
- [47] Santec. *Tunable Lasers*. website. Last retrieved: 12.04.2022. URL: https://www.santec.com/en/products/instruments/tunablelaser/?gclid=CjwKCAjwo8-SBhAlEiwAopc9W73BJQDgkCmkqCf4yhIcu_HwKbc9lw0PM2JpzJ-Tg-jZnDuS9qqjHB oCnEsQAvD_BwE.
- [48] Samuel Soldan, Gero Bonow, and Andreas Kroll. "RoboGasInspector - A Mobile Robotic System for Remote Leak Sensing and Localization in Large Industrial Environments: Overview and First Results". In: *IFAC Proceedings Volumes* 45.8 (2012). 1st IFAC Workshop on Automatic Control in Offshore Oil and Gas Production, pp. 33–38. ISSN: 1474-6670. DOI: <https://doi.org/10.3182/20120531-2-NO-4020.00005>. URL: <https://www.sciencedirect.com/science/article/pii/S147466701537244X>.
- [49] D. F. Swinehart. "The Beer-Lambert Law". In: *Journal of Chemical Education* 39.7 (1962), p. 333. DOI: 10.1021/ed039p333. eprint: <https://doi.org/10.1021/ed039p333>. URL: <https://doi.org/10.1021/ed039p333>.

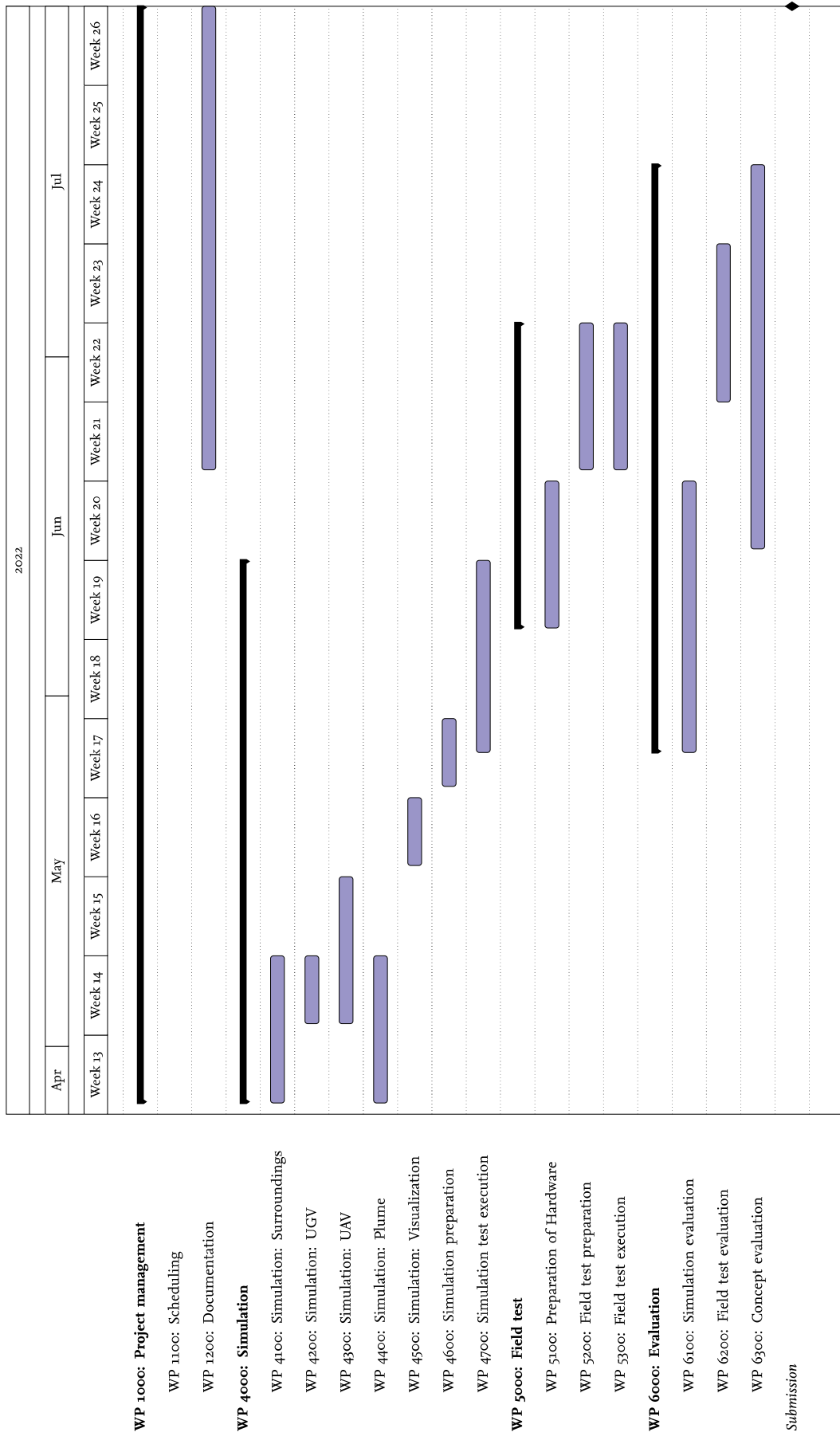
- [50] Unmanned Systems Technology. *Stabilizing Gimbals & Stabilized Camera Mounts for Drones & UAVs*. website. Last retrieved: 13.04.2022. URL: <https://www.unmannedsystemstechnology.com/company/gremsy/>.
- [51] *Traveling Salesperson Problem*. website. Last retrieved: 09.07.2022. URL: <https://developers.google.com/optimization/routing/tsp>.
- [52] Mike Wall. *The Most Extreme Human Spaceflight Records*. website. Last retrieved: 08.07.2022. URL: <https://www.space.com/11337-human-spaceflight-records-50th-anniversary.html>.
- [53] Fei Wang et al. "Recent Developments in Modulation Spectroscopy for Methane Detection Based on Tunable Diode Laser". In: *Applied Sciences* 9.14 (2019). ISSN: 2076-3417. DOI: 10.3390/app9142816. URL: <https://www.mdpi.com/2076-3417/9/14/2816>.
- [54] Thomas Wiedemann, Dmitriy Shutin, and Achim J. Lilienthal. "Model-based gas source localization strategy for a cooperative multi-robot system—A probabilistic approach and experimental validation incorporating physical knowledge and model uncertainties". In: *Robotics and Autonomous Systems* 118 (2019), pp. 66–79. ISSN: 0921-8890. DOI: <https://doi.org/10.1016/j.robot.2019.03.014>. URL: <https://www.sciencedirect.com/science/article/pii/S0921889018303816>.
- [55] Freddie Wilkinson. *The History of Space Exploration*. website. Last retrieved: 08.07.2022. URL: <https://education.nationalgeographic.org/resource/history-space-exploration>.
- [56] Dr. David R. Williams. *Luna 2*. website. Last retrieved: 08.07.2022. URL: <https://nssdc.gsfc.nasa.gov/nmc/spacecraft/display.action?id=1959-014A>.
- [57] Zaber. *X-G-RST-E Series: High load motorized gimbals with built-in controllers and motor encoders*. website. Last retrieved: 13.04.2022. URL: <https://www.zaber.com/products/gimbal-stages/X-G-RST-E>.

A. Project management

A.1. Work Breakdown Structure



A.2. Timetable



B. Storage Device

In the following a list of the content of the attached storage device is provided:

1. Digital version of the thesis
2. Latex code of the thesis
3. Result files of the Python simulation
4. Source code of the Python simulation
5. Source code of the Physics simulation
6. Literature