Domain Knowledge Assisted Robotic Exploration and Source Localization

Örebro Studies in Technology 89

THOMAS WIEDEMANN

Domain Knowledge Assisted Robotic Exploration and Source Localization

Title: Domain Knowledge Assisted Robotic Exploration
and Source Localization
Publisher: Örebro University 2020
www.oru.se/publikationer

Print: Örebro University, Repro 11/2020

ISSN 1650-8580
ISBN 978-91-7529-358-5

Abstract

Thomas Wiedemann (2020): Domain Knowledge Assisted Robotic Exploration and Source Localization. Örebro Studies in Technology 89.

Deploying mobile robots to explore hazardous environments provides an advantageous way to avoid threats for human operators. For example, in situations, where airborne toxic or explosive material is leaking, robots can be dispatched to localize the leaks. This thesis investigates a novel exploration strategy to automatically localize such emission sources with multiple mobile robots that are equipped with sensors to measure the concentration of the emitted gas.

The problem of localizing gas sources consists of two sub-problems that are addressed here. First, the thesis develops a method to estimate the source locations from sequences of localized concentration measurements. This approach can be also applied in case the measurements are collected by static sensor networks or human operators. Second, the thesis proposes an exploration strategy that guides mobile robots to informative measurement locations. With this strategy, a high level of autonomy is achieved and it is ensured that the collected measurements help to estimate the sources. As the main contribution, the proposed approach incorporates prior available domain knowledge about the gas dispersion process and the environment. Accordingly, the approach was coined Domain-knowledge Assisted Robotic Exploration and Source-localization (DARES). Domain knowledge is incorporated in two ways. First, the advection-diffusion Partial Differential Equation (PDE) provides a mathematical model of the gas dispersion process. A Bayesian interpretation of the PDE allows us to estimate the source distribution and to design the exploration strategy. Second, the additional assumption is exploited that the sources are sparsely distributed in the environment, even though we do not know their exact number. The Bayesian inference approach incorporates this assumption by means of a sparsity inducing prior.

Simulations and experiments show that the sparsity inducing prior helps to localize the sources based on fewer measurements compared to not exploiting the sparsity assumption. Further, the DARES approach results in efficient measurement patterns of the robots, which tend to start in downwind regions and move in upwind direction towards the sources where they cluster their measurements. It is remarkable that this behavior arises naturally without explicit instructions as a result of including domain knowledge and the proposed exploration strategy.

Keywords: Mobile Robot Olfaction, Robotic Exploration, Gas Dispersion Modelling, Bayesian Inference, Sparse Bayesian Learning

Thomas Wiedemann, School of Science and Technology Örebro University, SE-70182 Örebro, Sweden, thomas.wiedemann@dlr.de

Acknowledgements

First of all, my special thanks go to my supervisors Prof. Achim J. Lilienthal at the Örebro University and Dr. Dmitriy Shutin at the German Aerospace Center. I am very grateful for their support, help, and valuable feedback on my work, as well as their mentoring beyond scientific matters. I also want to acknowledge their time spend on reviewing our publications and their patience with my English language skills.

My gratitude also goes to the German Aerospace Center in general and in particular to the department of Communications Systems at the Institute of Communications and Navigation, for funding my Ph.D. studies. Further, I want to thank all my colleagues at the department for the great and friendly working atmosphere. Especially, I want to thank my colleagues of the Swarm Exploration group: (in order of appearance) Joachim Müller, Alberto Viseras, Christoph Manss, Valentina Karolj, Juan Marchal Gomez and Ban-Sok Shin. I would also like to add Emanuel Staudinger and Siwei Zhang to this list. I am grateful for the fruitful scientific discussions, their support in experiments and in general the great time we spent together. My special thanks also go to Monika Scheibe from the Institute of Atmospheric Physics. Together we conducted many field experiments even beyond the scope of this thesis and hopefully our cooperation will continue in the future.

Further, I am very grateful for the opportunity to stay for a few months at the Mobile Robotics and Olfaction Lab at AASS in Örebro. I enjoyed this research stay very much. I would like to thank all the people of the lab for their hospitality, for sharing ideas and scientific discussions and the great time. Namely, I would like to thank Han Fan, Dino Hüllmann, Ravi Teja Chadalavada, Victor Hernandez Bennetts and Erik Schaffernicht.

Last but not least, I want to thank my family and friends for their unconditional support throughout my Ph.D. studies.

Contents

List of Figures

List of Tables

Chapter 1
Introduction

1.1 Motivation: Applications for Model-based Multi-Robot Exploration

Nowadays most robots are employed in industrial applications, where they take over repetitive or physically hard tasks in order to relieve human operators. However, there is another important field of robotic applications with a long history, but different objectives compared to industrial applications: robotic exploration. In this field mobile robots are dispatched to explore, inspect or monitor an unknown area with the goal to gather information. For example, mobile robots already serve as emissaries of mankind on foreign planets and as our "eyes" in extraterrestrial exploration missions. Their story of success (and failures) reaches back to as early as 1970 when for the first time a lunar rover — Lunokhod 1 — was successfully sent to the moon by the Soviet Union [160]. Also for underwater and deep-sea exploration missions, mobile robots are the means of choice. A famous early example is the teleoperated underwater robot CURV-I developed by the U.S. Navy, which successfully recovered a lost H-bomb from the Mediterranean Sea in 1966 [169].

In general, mobile robots are very suitable when it comes to explore or operate in hazard and inaccessible environments. Most important, employing robots in such exploration missions has the distinct advantage that threats for human operators are avoided. This does not only apply to extraterrestrial or deep-sea missions. It also pertains to Chemical, Biological, Radiological and Nuclear (CBRN) events and disaster relief scenarios where an environment to be explored is contaminated by toxic or dangerous material. In such disaster relief scenarios, robots are required to explore the environment as fast as possible and provide reliable information early enough to civil protection agencies and first responders. Here, the required time to explore a region can make the difference between life and death.

In this respect, multiple robots are superior compared to a single robot. Multiple robots can observe different locations at the same time and can distribute the workload among each other. In this way a multi-robot system can accelerate the required exploration time. Moreover, a multi-robot system contains natural redundancies which make the whole system more robust, since a failure of an individual robot can be compensated by others.

To the present day, most of the robots employed in exploration missions are teleoperated [48, 58, 156]. This means that a human operator remotely controls the robot, while the robot provides measured information about its surrounding, like pictures or videos to the operator. Whereas a single robot might be easily controlled by a single, skilled operator, steering and coordinating many robots in real time is far too complex for a single operator. Even for a team of operators coordinating a large number of robots would be challenging. Moreover, in disaster scenarios human resources taking care of controlling multiple robots might be scarce. Thus, a high level of autonomy of the robots is desirable in order to coordinate and control the multi-robot system independently of an operator. The robots should decide on their own where to go and how to coordinate with other robots, e.g. to avoid collisions. They should efficiently explore the environment in order to collect as much information as possible and as fast as possible. The design of such an intelligent exploration strategy guiding the robots is the topic of this thesis.

Unfortunately, robots have two major drawbacks compared to a human operated exploration mission. First, compared to human operators, robots lack of curiosity. Curiosity can be defined as a strong desire to know more about something, which is the essential motivation driving exploration missions. For an exploration strategy it is thus fundamental to develop an “artificial curiosity”, which guides and drives the robots. In practice, this thesis will show how to quantify the knowledge and uncertainty about the exploration environment. By successively reducing the uncertainty about the environment based on new observations, the proposed exploration strategy will mimic a behavior as if the robots are being driven by their curiosity.

Second, human operators typically contribute their expert knowledge — which the robot does not have — to the exploration mission. Human operators often possess a deeper understanding of what is to be expected in the exploration environment. Their understanding will help a-priori to distinguish informative locations from less informative ones, which makes the exploration more efficient. In order to accomplish a similar behavior of the autonomous robots, this thesis proposes to assist the robots with a-priori available domain knowledge. The domain knowledge will be encoded in a mathematical model describing the environment surrounding the robots. This thesis will show how such a model-based exploration strategy for a multi-robot system can be designed and how it performs.

In general, the presented concept of model-based exploration for a multi-robot system can be applied to a wide variety of exploration tasks. Nevertheless, this thesis focuses on one particular application: the problem of exploring a gas distribution in order to localize gas sources.

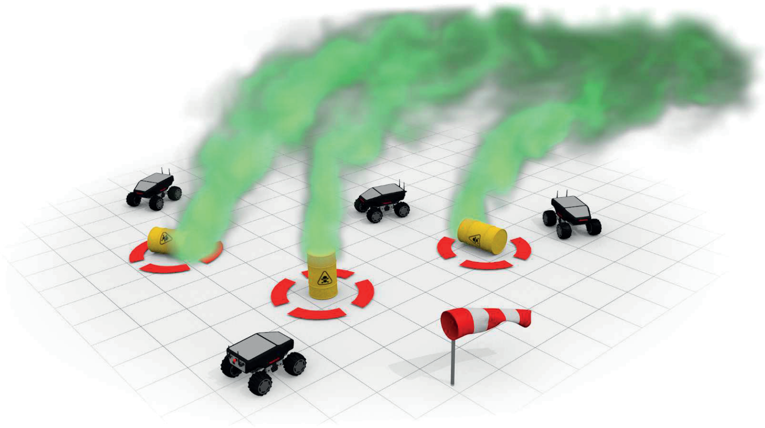


Figure 1.1: The thesis tackles the problem of finding gas sources, such as those caused by a technical accident, using multiple robots. The robots are able to measure the gas concentration as well as other environmental parameters, like wind, at their current position.

1.2 Challenge: Gas Source Localization

As a dangerous implication of many CBRN events or technical accidents, toxic or dangerous material could leak and spread as airborne trace substances. In such scenarios it is important to localize the leaks in order to fix them. Also in other applications it is of interest to find sources of airborne trace substances. For example in geophysics volcanic emission sources, e.g. SO_x, can be explored by a flying robot [117]. Localizing methane leakage from landfill sides is another application [13, 28].

Likewise, this thesis addresses the gas source localization problem by means of a multi-robot system. More precisely, we deal with the following setup: We consider a spatially bounded region, which we refer to as the exploration environment. We assume that multiple gas emission sources are spatially distributed in the region. These sources emit a certain invisible, airborne gas, which is spreading over the whole environment. We do not know the number of sources; neither do we know their locations nor their strength. However, our goal is to find and localize these sources.

To this end, multiple mobile robots are dispatched. The robots are equipped with sensors in order to measure environmental parameters. More specifically, the robots can measure the gas concentration at their current position. Other environmental parameters, like wind, are considered to be either known or measurable, too. It is important to note that the robots cannot directly measure a emission source. They can only measure concentrations and based on their measurements infer information about sources. Correspondingly, the gas source localization in the context of this thesis can be considered as an inference problem. As such, the first challenge tackled in this thesis is:

> ! **Infer the spatial source distribution based on concentration measurements taken by the robots!**

As a second challenge it has to be decided where to take the measurements. We conjecture that by sampling the gas concentration in an intelligent way, the inference problem may be solved more easily. Or looking at it the other way round: if the gas concentration measurements are taken at disadvantageous locations, it might be difficult or even impossible to infer the spatial source distribution. Therefore, the second challenge considered in this thesis is:

> ! **Design an intelligent sampling strategy for a multi-robot system!**

The exploration strategy needs to determine informative measurement locations for the multi-robot system. In our context an informative location implies that a measurement at this location facilitates the inference of the sources. By providing these informative locations as way points for the multi-robots system, the exploration strategy actually guides the robots.

In general, the two outlined challenges — inferring the source distribution and designing an exploration strategy — might be treated separately. However, it is the nature of the beast that the two problems are tightly coupled. Thus, in this thesis we choose a holistic point of view and handle the two problems as one.

It is important to note that while this section only provided a rough overview about the problem tackled in this thesis, a formal and more detailed formulation of the problem will be provided later in Chapter 3 and 4.

1.3 Contribution: Model-based Exploration Strategy

As already indicated in the motivation, one of the main contributions of this thesis is to assist the source localization strategy by domain knowledge. In the thesis we will refer to this approach as the Domain-knowledge Assisted Robotic Exploration and Source-localization (DARES) strategy, which also leads to the first research question of the thesis:

> ? **How to incorporate domain knowledge from physics into an exploration strategy for gas source localization?**

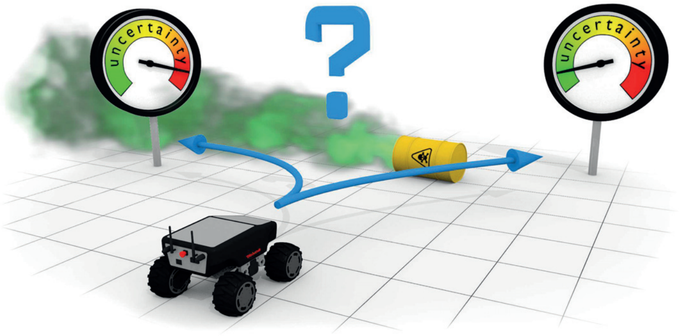


Figure 1.2: The DARES approach sends robots to locations where the uncertainty of the gas dispersion process according to the model is currently high.

In particular, we would like to incorporate knowledge about the physical behavior of gas dispersion. Our approach is to make use of a mathematical model that describes the physics of gas dispersion. The use of such a mathematical model turned out to have three distinct advantages:

1. An appropriate gas dispersion model enables us to infer the sources driving the gas dispersion. The model sets up a mathematical relation between the gas concentration distribution in the environment and the spatial distribution of sources. Essentially, the gas sources can be considered as parameters of the gas dispersion process. In this way the gas source localization problem turns into a parameter estimation problem. In other words, our inference problem boils down to calculating the spatial source distribution based on the gas dispersion model and gas concentration measurements.

2. Adding domain knowledge to the exploration strategy in the form of a gas dispersion model speeds up the gas source localization. While this argument may seem speculative at this stage, the experimental evaluation presented further down in this thesis will show that it is actually true. Loosely speaking: the advantage comes from the capability of the model to interpolate in between measurement locations correctly (in a domain-knowledge informed way). Thus, there is no need to sample the exploration environment everywhere and the sources can be localized with fewer measurements compared to a model-free exploration approach.

3. Based on a mathematical gas dispersion model it is not only possible to estimate the sources, but also to quantify the uncertainty of these estimates at different locations in the exploration environment. Uncertainty quan-

tification is in general a very useful information-theoretical tool. Here, it is used to build a gauge for information available about certain locations in the environment. By means of this gauge, the exploration strategy can propose locations with currently high uncertainty, i.e. low information.

It is important to remark that the DARES strategy proposed here is adaptive (sometimes also called reactive). In contrast to a predefined strategy that follows a plan or trajectory that was defined beforehand, an adaptive strategy reacts to current measurements and adjusts its plans accordingly [98]. This implies that the processing of measurements and the calculations of informative locations has to be done online and in real-time.

By having a closer look at other gas source localization approaches in the literature, it becomes apparent that the usage of a gas dispersion model is not uncommon. In fact the DARES approach proposed here can be categorized as an infotactic source localization strategy. (Section 6.1.2 will give a more detailed explanation of infotaxis.) The phrase infotaxis is an umbrella term for all information or entropy-driven source localization approaches [180, 70, 126, 152]. As such, all these approaches require a mathematical model in order to define and calculate information content, or entropy.

However, model-based approaches have a categorical disadvantage: The model might be wrong! Essentially, all models only approximate reality. Therefore, there is always a mismatch between the model's predication and the real phenomenon. This mismatch is actually crucial in case a mathematical model makes use of strong simplifications of the real phenomenon. Strong simplifications are often unavoidable and are the necessary price for keeping the computational complexity of the model reasonably low. In our case, the real gas dispersion process is a very complex phenomenon. The mathematical model, however, has to be simple enough for online calculations required by the DARES approach. Hence, here we have to face strong model mismatch effects.

Paradoxically, when making use of simplifications, most scientists and engineers are aware of the mismatch and limitations of their model. Nonetheless, often they cannot exploit their knowledge about limitations. This circumstance leads to the second research question of this thesis:

> ? **How can we quantify the trust into our model and take it into account in the exploration strategy?**

To address this question, we propose to embed the gas dispersion model in a probabilistic framework. While a classical deterministic model sets up hard mathematical constraints, in a probabilistic framework we can relax and soften model assumptions. We will even show how to parametrize and quantify the relaxation of model assumptions. With this parametrized relaxation it becomes possible to express our trust in the model. Experimental evaluations carried out in this thesis will analyze the effect of the relaxation on the exploration performance. By taking into account the imperfection of the model and model mismatch effects, we obtain an overall more robust system even with a simple gas dispersion model.

State-of-the-art model-based gas source localization strategies known from literature can handle either only a single gas source or an a-priori known number of sources [152, 51, 24, 11]. Knowing the number of sources may be an unrealistic assumption, especially in disaster scenarios. As another contribution, this thesis aims to avoid assumptions about the number of sources, which leads to the third research question:

> **How can we avoid assumptions regarding the number of sources?**

In general not knowing the number of sources makes the problem of estimating the source distribution based on concentration measurements extremely complex. While for a single source it is sufficient to estimate a single location and the source strength, the solution space for an unknown number of sources is infinitely large. A large solution space is particularly problematic in the context of exploration, since only few measurements are available to support the estimation (at least at the initial phase of the exploration). However, for non-artificial applications it is justified to assume that the number of sources is not infinite. Actually, for many applications we can even assume that there are only a few sources, although we do not know their exact number. Therefore, in this thesis we will make use of the assumption that the sources are only sparsely distributed in the environment. By designing an appropriate prior for the probabilistic model, we take this additional information into account. As we will show, it does not only regularize the source estimation problem, but also turns out to be very beneficial for the performance of the proposed DARES strategy.

Finally, besides answering the three research questions mentioned above, the thesis also aims at answering the following question:

> **How to take advantage of the multi-robot system in the implementation?**

In fact, the multiple robots with their on-board computers and a communication system can be considered as a mobile computer cluster. By using distributed algorithms, we can profit from this architecture. Thus, thesis will also provide a novel way to implement the required calculations in a distributed fashion. In other words, the robots can share the computational costs of the algorithm. To this end, the probabilistic gas dispersion model is translated into a factor graph [88] representation. Based on the factor graph, a message passing algorithm is developed, which is extremely suitable for a distributed implementation.

1.4 Terminology

Robotic gas source localization as tackled in this thesis is challenging because of its interdisciplinary character. As we will see, the used methods vary from variational calculus over numerical analysis and information theory to robotics. To avoid misunderstandings and to take various backgrounds of readers into account, the following list will define the most important phrases and terminology.

Robot The thesis refers to a robot as a wheeled mobile robotic platform capable of moving around in the exploration environment and carrying sensors. We further assume that robots are equipped with an on-board computer to carry out calculations required by the exploration strategy. The term agent might be used synonymously in literature.

Multi-Robot System denotes the union of multiple robots. In this work we will consider less than 10 robots. To make the system functional we assume that the robots are able to share information among each other by means of a communication system. As a synonym one may find multi-agent system or robotic swarm in literature.

Environment The thesis refers to the term environment to describe the spatial area surrounding the robots that has to be explored. This region is bounded and its boundaries are a-priori known. In the mathematical derivation we will also refer to the environment as spatial domain.

Process is always understood as a gas dispersion process in this thesis. It denotes the dynamic phenomenon of gas emitted from sources spreading out into the environment. To be precise: it is actually this process that has to be explored and not the environment itself, where the process takes place.

Model In this thesis a model is a mathematical description of the gas dispersion process. The model is used to encode domain knowledge a-priori available and to assist the exploration.

Exploration Strategy In the context of gas source localization, exploration means to take observations of the gas dispersion process, and to estimate the sources based these observations. The exploration strategy denotes the algorithmic procedure where and when the observations should be taken by the robots. As a key contribution this thesis proposes the Domain-knowledge Assisted Robotic Exploration and Source-localization (DARES) strategy. It is an reactive strategy which means that the location of the next observation depends on the previously taken observations.

Distribution In this thesis, the word distribution occurs in two different contexts, which may lead to some misunderstanding. In one context it is used for spatial distributions like the spatial source distribution describing the placement of the sources or the distribution of gas in the environment. The word distribution is also used in a probabilistic context to describe Probability Density Functions (PDFs) of random variables. These two different concepts can be, however, distinguished from their context.

Sparsity It is assumed that the sources are sparsely distributed in the environment. This means that only a few sources are located within the exploration area. It does not require that the sources are far away from each other. The assumption only concerns their number. Later on we will represent the source distribution by a vector. Then the sparse source distribution is reflected by the fact that this vector is sparse, i.e. most elements are zero.

Ground Truth data describe the actual situation, e.g. the actual source distribution. In the evaluation part of this thesis we compare the estimates, e.g. estimated source distribution, to the ground truth data. In a real application, ground truth data are not always available; however they are essential in experiments for a scientific evaluation.

1.5 Publications

A long with this thesis, several journals and conference publications have emerged from the research work presented here. The following list summarizes the most important publications that directly contributed to the thesis.

> Wiedemann, T.; Manss, C.; Shutin, D.; **Multi-agent exploration of spatial dynamical processes under sparsity constraints;** Journal of Autonomous Agents and Multi-Agent Systems (JAAMAS), 2018 [189]
>
> This article presented model-based exploration strategy for the first time (submitted 2016). However, the model used [189] in had very strong simplifications and limitations. Nonetheless, the factor graph representation and message passing algorithm shown in Section 5.1 of the thesis were first developed in this article.

Wiedemann, T.; Manss, C.; Shutin, D.; Lilienthal, A.J.; Karolj, V.; Viseras, A.; **Probabilistic modeling of gas diffusion with partial differential equations for multi-robot exploration and gas source localization;** European Conference on Mobile Robots (ECMR), 2017 [190]

This paper implemented exploration strategy presented in this thesis based on a simplified gas dispersion model in Hardware-in-the-Loop experiments.

Wiedemann T.; Shutin D.; Lilienthal A.J.; **Model-based gas source localization strategy for a cooperative multi-robot systems - A probabilistic approach and experimental validation incorporating physical knowledge and model uncertainties;** Journal of Robotics and Autonomous Systems (RAS), 2019 [192]

This article is an extension of the ECMR 2017 paper [190]. Still based on a simplified model, the exploration strategy was evaluated in Hardware-in-the-Loop experiments, but also in real-world experiments. The results of these experiments are also shown in Chapter 7 of this thesis.

Wiedemann, T.; Shutin, D.; Hernandez, V.; Schaffernicht, E.; Lilienthal, A.J.; **Bayesian gas source localization and exploration with a multi-robot system using partial differential equation based modeling;** ISOCS/IEEE International Symposium on Olfaction and Electronic Nose (ISOEN), 2017 [191]

This paper showed the capability of the presented exploration strategy in the presence of model mismatch effects. To this end Hardware-in-the-Loop experiments with a filament based gas simulator were carried out.

Wiedemann, T.; Lilienthal, A.J.; Shutin, D.; **Analysis of Model Mismatch Effects for a Model-Based Gas Source Localization Strategy Incorporating Advection Knowledge;** Sensors, 2019 [188]

This article introduced a more complex gas dispersion model for the exploration strategy. This model and its numerical and probabilistic treatment are explained in Chapter 3 and 5 of this thesis. Also the variational inference approach developed in the article is presented in Section 5.2. Moreover, the article presented an extensive parameter evaluation in simulations. Section 7.1 reproduces these parameter studies.

1.6 Outline

The rest of the thesis is structured as follows:

Chapter 2 The next chapter provides information on technologies and state-of-the-art of robotic gas sensing. It explains operating principles of common gas sensors suitable to be carried by a robot and robots suitable to carry as gas sensor. In addition, different objectives of robotic gas sensing related to this thesis are presented.

Chapter 3 selects an appropriate gas dispersion model, after presenting the most relevant options found in literature. Namely, the thesis chooses the advection-diffusion Partial Differential Equation (PDE) to mathematically describe the dispersion process. The chapter explains how we can handle the Partial Differential Equation (PDE) numerically. This results in a system of algebraic equations as a final description of the gas dispersion process. The chapter also presents an observation model representing the sensor characteristics of the gas sensors carried by the multiple robots. In addition, in this chapter we comment on the importance of a proper regularization in order to estimate the source distribution based on gas concentration measurements. There, the assumption of sparsely distributed sources plays an important role. Finally, an example shows how to estimate the source distribution based on measurements, the presented deterministic model and the sparsity assumption.

Chapter 4 translates the deterministic model developed in Chapter 3 into a probabilistic form. In particular, a Bayesian approach is chosen. Thereby, the novel ideas of relaxing the model assumptions and parametrizing our trust into the model are introduced. Further, the chapter shows how to represent the assumption of sparsely distributed sources in the probabilistic framework. This is achieved by Sparse Bayesian Learning (SBL) techniques and designing an appropriate sparsity inducing prior for the Bayesian inference approach. Last but not least, a probabilistic airflow model is presented to incorporate wind information into the probabilistic framework.

Chapter 5 is divided into two parts. While the previous chapters only focused on formulating the estimation problem, the two parts of this chapter show how to actually solve the problem, i.e. estimating the source distribution (and gas concentration distribution). The first part tackles a special case of the gas dispersion model, where the system of algebraic equations is linear. For this case the probabilistic model is cast into a factor graph. Based on the factor graph, a message passing algorithm is developed suitable for a distributed implementation on the multi-robot system. The performance of the message passing algorithm is compared to other state-of-the-art solvers. The second part shows how to implement a solver for the general, non-linear gas dispersion model. To this end, a variational inference approach is presented.

Chapter 6 develops the actual exploration strategy for a multi-robot system. It first gives an overview on other common gas source localization approaches. Then, the chapter explains how to make use of the probabilistic framework presented in Chapter 4 and the implementation of Chapter 5 in order to design the Domain-knowledge Assisted Robotic Exploration and Source-localization (DARES) strategy for a multi-robot system. Further, the chapter shows how multiple robots can be coordinated in order to follow the exploration strategy without colliding with each other.

Chapter 7 evaluates the proposed DARES approach. In particular, it is evaluated how well robots are able to localize the sources, i.e. estimate the source distribution. Therefore, numerous experiments are presented. The DARES strategy is first tested in pure simulations. The simulation setup enables an extensive study of parameters used to regulate the exploration strategy. In a second step, the DARES approach is evaluated in Hardware-in-the-Loop experiments, where the gas dispersion is still simulated, but a real multi-robots system is employed. These experiments reveal how the performance of the exploration strategy is affected by real-world constraints like the dynamics of the robots or communication constraints. Beside the gas source localization performance, the chapter also shows the capability of the air flow model used in the experiments. In a last step, the airflow model is incorporated into the exploration strategy. Then, the whole system is finally tested in real-world experiments with ethanol.

Chapter 8 summarizes and discusses the thesis and its contributions. It further suggests future research directions that arise from this thesis.

Chapter 2
Background on Robotic Gas Sensing

This chapter provides a general overview on technology, state-of-the-art and research fields related to gas source localization and gas sensing robots. First, it will explain fundamentals of gas sensing and operating principles of state-of-the-art gas sensors. Then, robotic platforms typically employed for gas sensing and Mobile Robot Olfaction (MRO) tasks are summarized. Finally, different objectives of robotic gas sensing are explained as well as research work carried out in these fields.

2.1 Sensor Technologies

When it comes to explore gas distributions, monitor gas concentrations or search for gas leaks with robots, gas sensors are essential. Gas sensors, especially sets of gas sensors, are often considered as an artificial, electronic nose of the robot [96], [115] A large research field with a long history is dedicated to the development of such electronic noses, see e.g. [136], [65]. The following text will give a short overview of available sensor technologies which are suitable to be carried by a robot and have been used in robotic applications. These are sensing technologies we considered for this thesis as well.

2.1.1 Electrochemical Cell

An electrochemical cell consists basically of two electrodes: the sensing electrode (sometimes called working electrode) and the counter electrode. Both electrodes are in touch with an electrolyte, e.g. sulfuric acid [15]. Gas, which comes in contact with the sensor, diffuses to the surface of the sensing electrode, where either an oxidation (for CO, H_2S, NO, SO_2) or reduction (NO_2, Cl_2) occurs [15]. This reaction

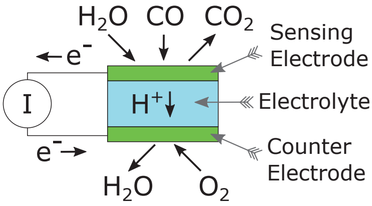


Figure 2.1: Operating principle of a Electrochemical Cell (adapted from [8])

is catalyzed by the material and surface properties of the electrode [43]. In figure 2.1 the working principle is shown for carbon-monoxide (CO) as an example. On the sensing electrode the carbon-monoxide is oxidized to carbon-dioxide (CO_2) by means of water contained in the air [15].

$$H_2O + CO \rightarrow CO_2 + 2e^- + 2H^+ \tag{2.1}$$

The resulting protons (H^+) travels through the electrolyte to the counter electrode, while the resulting electrons causes a measurable current flowing from the sensing electrode to the counter electrode. On the counter electrode a reduction of oxygen takes place [15]:

$$\frac{1}{2}O_2 + 2e^- + 2H^+ \rightarrow H_2O \tag{2.2}$$

Since a current is generated, this type of sensor is also referred to as an amperometric gas sensor or a micro fuel cell [43]. In practice, most electrochemical gas sensors possess a third electrode (reference electrode). The reference electrode is used to stabilize the potential of the sensing electrode. A stable and constant potential of the sensing electrode is important to achieve a constant sensitivity, good linearity and minimum sensitivity to other interfering gases [15]. Electrochemical cells are light and can be carried by small robotic platforms, e.g. Unmanned Aerial Vehicles (UAVs) [117], especially for the purpose of air pollution monitoring [183].

Target Gases: CO, H_2S, NO, SO_2,NO_2, Cl_2
Advantages: low power consumption, high sensitivity, light-weight
Disadvantages: long response and recovery times (typically $>$ 10 seconds), cross sensitivity

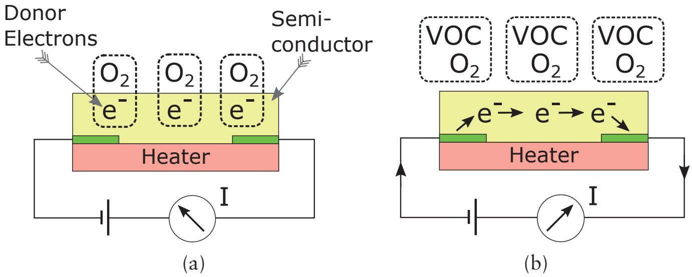


Figure 2.2: Operating principle of a MOX sensor (adapted from [8])

2.1.2 Metal Oxide (MOX) Gas Sensors

The measuring principle of MOX gas sensors (also called Metal Oxide Semiconductor (MOS)), was first proposed in [157]. The sensor consists of two electrodes as depicted in Figure 2.2. The two electrodes are connected by a thin semiconductor layer, such as oxides of tin or zinc (SnO_2, or ZnO) [187]. These n-type materials act as a donor of electrons. When heated up, the surface layer of the semiconductor absorbs oxygen of clean air. The oxygen attracts the donor electrons at the surface layer and the resistant of the semiconductor increases (see Figure 2.2a). When exposed to reducing gases e.g. Volatile Organic Compounds (VOC), the surface layer also absorbs these gases and the gas reacts with the already absorbed oxygen. As a consequence the bound electrons are released and the resistant of the semiconductor decreases (see Figure 2.2b) [187]. Thus, in a MOX sensor the surface conductivity of semiconductors changes based on reactions of gases with atmospheric oxygen that is already absorbed by the semiconductor's surface layer [187]. This change of conductivity can be measured and is related to the concentration of the gas. MOX sensors are very popular in robotic applications and can be found in many publications [103, 28, 185, 115, 118, 101].

Target Gases: CO, VOC, CH_4
Advantages: high sensitivity, light-weight, inexpensive
Disadvantages: drift, cross sensitivity

2.1.3 Photoionization detector (PID)

The main component of a PID sensor is a short-wavelength ultra-violet lamp. This lamp emits high energy photons which are entering through a crystal window a chamber that contains gas samples [142]. Commonly, the gas can freely diffuse into the chamber through a membrane filter [14]. In the chamber VOC molecules, are ionized by the emitted photons. This results in positively charged ions and ejected electrons [142]. The chamber is surrounded by an anode and cathode as indicated in Figure 2.3. These electrodes cause an electrical field in the chamber that attracts the positive charged ions and electrons [14]. The resulting current can be measured and is proportional to VOC concentration. Note that the photoionisation effect only takes place, if the ionization energy of the the VOC is below the energy of the photons. The

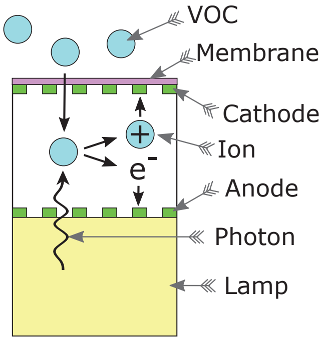


Figure 2.3: Operating principle of a PID sensor (adapted from [142])

energy of the photons, however, depends on the used lamp. While an Argon lamp with 11.7 eV can detect the largest range of VOC, a Xenon lamp with 9.6 eV has the highest selectivity [14]. The most common lamp, however, is a Krypton lamp that emits photons with 10.6 eV. Because of their fast response time and long term stability, PIDs are common in robotic applications [29, 124].

Target Gases: most VOC
Advantages: high sensitivity, fast response, linear characteristics
Disadvantages: price, limited target gases

2.1.4 Tunable Diode Laser Absorption Spectroscopy (TDLAS)

TDLAS gas sensors work based on the spectroscopy principle. Every molecule absorbs light-energy in very specific narrow bands around certain wavelengths in the electromagnetic spectrum [63]. The wavelengths are specific for different types of molecules [93]. This effect is used in the TDLAS sensing principle. A TDLAS sensor consists of a laser with a wavelength tuned to the specific wavelengths of a target gas. When the laser beam transverses a plume of the target gas, the gas attenuates the laser energy [63]. As shown in Figure 2.4, the laser beam needs to be scattered back by any surface in the environment. The returning light is then analyzed by a photodetector in the TDLAS sensor. Based on the absorbed energy for a specific wavelength, the gas concentration along the light path can be inferred. Essentially, the sensor provides a concentration value integrated along the light path. TDLAS can be categorized as a remote sensor, since the sensor itself does not need to be inside the gas plume. In contrast, the in-situ sensors listed above only provide a point-wise gas concentration measurement and need to be inside the gas plume. A TDLAS sensor has been used on robotic platforms (including UAVs) for gas sensing tasks [31, 20, 132]. However, the processing of the collected measurements is more involved [20], as compared to other sensors.

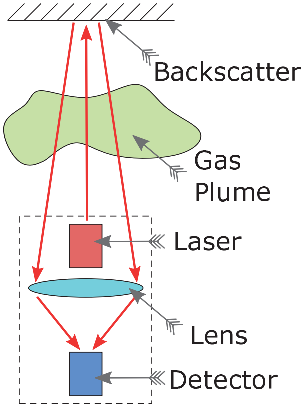


Figure 2.4: Operating principle of a TDLAS sensor (adapted from [63])

Target Gases: commonly CH_4 (but not limited to)
Advantages: high selectivity, fast response, remote sensing
Disadvantages: price, weight

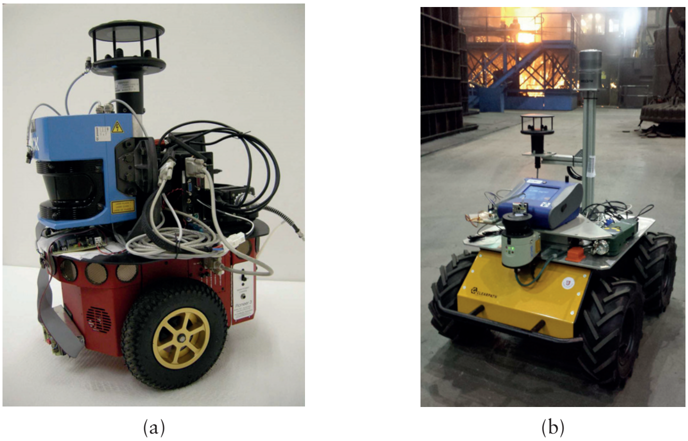
2.2 Robotic Platforms

Besides a gas sensor, a robotic platform to carry the gas sensors is required for robotic gas sensing tasks. The following section gives an overview about common state-of-the-art robots and robotic platforms dedicated to carry gas sensors. Two main classes – ground based robots and flying platforms – are discussed with respect to their advantages and disadvantages. As a remark, measuring water pollution is quite similar to airborne gas sensing. In those cases autonomous surface vehicle (vessels) or unmanned underwater vehicles are employed, e.g. [21]. However, such platforms are not part of the following overview.

2.2.1 Ground Based Robots

For the purpose of this work, as a "ground based robot" we typically understand a mobile robotic platforms that are driven by wheels, tracks or legs. However, their action space is limited to two dimensional movements. Ground based robots are very popular for gas sensing tasks, because of their high payload capability and long operation time [28]. For sensing gases heavier than air, their limitation to move on the ground is noncritical. For gases lighter than air which do not stay close to the ground, the limited action space might be a problem.

When the research field of robotic gas sensing and MRO started to emerge in the 1990s, most research work was focused on small scale ground based robots [154], [86]. Over time bigger ground based robotic platforms are em-



(a)

(b)

Figure 2.5: Two examples of a ground based robot: The robots depicted in (a) was for example used in [145] and the robots shown in (b) in [31] (photograph courtesy of Han Fan and colleagues).

ployed, since they are capable of carrying multiple gas sensors and additional sensors like anemometers or laser scanners to measure environmental parameters or to navigate [144], [118], [31]. A higher payload also allows carrying more powerful computers for advanced on-board data processing and decision making. It is remarkable that most gas sensing robots are developed solely for research purpose, and only few commercial robots are available dedicated to gas sensing tasks (as an exception see e.g. SMP Robotics Systems Corp. [9]).

When it comes to control ground based robots for gas sensing tasks, recently there is a clear trend towards advanced and sophisticated planning and control algorithms. In the past the robot's motor controller was often directly coupled to the gas sensors as for example by the Braitenberg vehicle [34], [99]. Nowadays, robots make use of path planning algorithms that allow moving without collisions in complex outdoor [31], [144] or indoor environments [29], [118], [125], [20].

2.2.2 Flying Robotic Platforms

Flying robotic platforms – UAVs – are of particular interest for gas sensing tasks, since they are able to move in three dimensions. Thus, they can be employed to generate 3D gas distribution maps [111]. Nevertheless, they are sometimes still constrained to operate in a 2D space on a constant height over ground [130], [129]. Even in these cases they have an advantage compared to ground based robots, because of their higher mobility and fewer restrictions by obstacles or rough terrain.

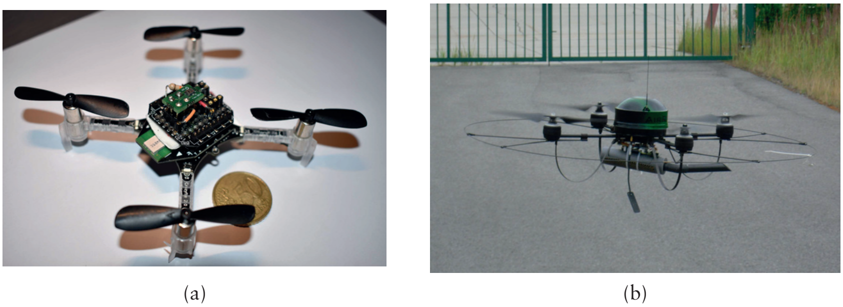
There are three main types of small scale UAVs operating in gas sensing tasks: fixed wing UAVs, rotary wing UAVs and blimps.

Fixed wing UAVs

In general a fixed wing UAV has a longer endurance and flies at higher speeds compared to rotary wing UAVs. However, they lack of hovering capability, which is often important for accurate in-situ measurements [110] where the sensor needs to stay at a location for a certain time. Still, fixed wing UAVs are sometimes employed for air quality monitoring tasks; for example to measure the carbon dioxide [186] or particulate matter concentration [73], [18]. In [128] a fixed wing UAV is even used to quantify methane leakage from a natural gas compressor station.

Rotary wing UAVs

Due to their hovering capability, rotary wing UAVs are more popular for gas sensing tasks. Especially modern multi-copter platforms with an electric propulsion system are often used to carry gas sensors [129], [131], [168], [80], [164], [76]. Also traditional unmanned helicopters with combustion engines can be



(a)

(b)

Figure 2.6: The picture in (a) shows the nano rotary wing UAV as used in [36] carrying a MOX sensor (photograph courtesy of Javier Burgués). In (b) a UAV is shown as used in [130] (photograph courtesy of Patrick Neumann).

used for gas sensing applications. For example in [117] a helicopter platform was used to measure the carbon dioxide flux at a volcano. Again air quality monitoring is a common task for rotary wing UAVs [17], [182], [55]. In some approaches like [41], the UAV does not directly carry a gas sensor. Instead, it transports a sampling device to collect a gas sample stored in a canister that is brought back to be analyzed in a laboratory later on.

When employing rotary wing UAVs for gas sensing, it is important to be aware of the interaction of the UAV's propulsion system with the gas or plume. The down wash of the propulsion system disturbs and destroys the natural structure of a gas plume and has to be considered. Studying the effect of the down wash of multi-copters is therefore an active research field [54], [110], [197], [45]. The impact of the down wash is mostly dependent on the size and weight of the UAV. This fact makes nano UAVs particularly interesting, because of their small size and weight and thus their modest interaction with the gas dispersion. For example in [36] a nano UAV with a total weight of about 30g carrying a MOX sensor was employed for gas mapping and source localization.

Blimps and balloons

The use of a third type of UAVs can reduce the impact of the propulsion system even further: for small scale blimps (airships) and balloons the effect can be neglected. Furthermore, they possess a longer endurance compared to fixed or rotary wing UAVs. Unfortunately, they are strongly affected by weather conditions, such as wind [182], and show a bad size-to-payload proportion [130] compared to other UAVs. Nevertheless, small scale blimps have been employed in gas sensing tasks, for example in [84] or [140] to create a 3D gas distribution map in an indoor environment. Tethered balloons on the other hand are very common for atmospheric measurements, e.g. [62], and can carry heavy gas sensing equipment compared to small UAVs. However, their limited mobility disqualifies them for most robotic gas sensing tasks.

The approach presented in this thesis relies on a model of the gas dispersion process based on physics. The interaction of an UAV's propulsion system with the gas dispersion is a very complex phenomenon and could over complicate the analysis of the presented approach. Therefore, in this thesis we stick to ground based robots as a platform for carrying gas sensors.

2.3 Objectives of Robotic Gas Sensing

Robots are employed in gas sensing applications with different objectives. One of the objectives we follow in this work is localizing gas sources. Apart from gas source localization, the following section provides a short overview on other objectives commonly found in literature and their relation to the approach proposed in this thesis.

2.3.1 Gas Source Localization

Robotic gas source localization aims at finding (a) gas source(s) based on gas concentration measurements taken by sensors mounted on a robotic platforms. Often additional environmental parameters are measured to support localizing the source, such as wind [144]. Even visual information can help to identify potential source candidates, as in [123]. However, the main practical challenge to find gas sources is the fast decrease of gas concentration with increasing distance from the source [86]. Thus, the gas plume emitted from the sources shows a diluted structure further away from the source [86], which makes detection and localization challenging.

According to [75] the source localization task can be divided into three subtasks: first finding the gas plume, then tracing the plume back to the source (often called plume tracking) and, finally, declaring the source as found. Most of the research is focused on the first two steps, while the declaration of a source is often neglected or assigned to a human supervisor. The holistic approach presented in this thesis will cover all three points in a single framework. It will be shown that the model-based uncertainty-driven exploration strategy will just uniformly sample the environment as long as no gas concentration is measured. As soon as a gas concentration is measured, i.e. the plume is detected, the presented approach will automatically concentrate the samples in areas with higher gas concentration. This can be considered as plume tracking. Further, due to the use of a gas dispersion model the presented approach is able to infere where the sources are, based on the concentration measurements.

In general, it is difficult to compare different source localization approaches regarding their performance, since they employ different robots, different sensors and consider different environments [86]. In the past, many source localization strategies were inspired by nature [100], [106], [130], [64], e.g. the behavior of moth, lobsters etc. A more detailed description of these bio-inspired strategies will be provided in Section 6.1.1. However, these approaches show significant limitations. Often, they rely on gradients of the gas concentration and assume smooth gas concentration distributions. In reality, the gas concentration is often patchy and disturbed by turbulence, which results in poor performance of these approaches [133]. In general, bio-inspired gas source localization strategies have to be carefully adapted to robotic applications as discussed in [28],.

With respect to the limitation of bio-inspired approaches, more sophisticated gas source localization strategies have been developed. For example, infotaxis is a quite popular gas source localization strategy in robotics nowadays [70], [152], [118], [126]. Briefly speaking, these approaches try to maximize information about the gas sources. Section 6.1.2 will give a more detailed explanation of infotaxis. Also, the exploration strategy proposed in this thesis can be considered as an infotactic approach. Furthermore, we employ multiple robots as in [118], [163], [69], [70], [33], instead of a single robot as in early approaches of robotic gas source localization [82], [83]. Even though multiple robots are employed, most approaches consider only a single source, e.g. [118], [163], [69], [70]. In this respect the DARES approach of this thesis particularly contributes a gas source localization strategy that is capable to localize multiple sources and does not require knowing the number of sources a-priori.

2.3.2 Plume Tracking

As mentioned above, plume tracking (also called plume tracing) is actually a sub-task of gas source localization [75]. However, many research publications focus solely on this topic. Often the core idea of plume tracking by means of a mobile robot is to stay in the gas plume and move in upwind direction towards the gas emitting source. This strategy is grounded on the fact that wind or air flow is the main transportation mechanism of airborne gas [99]. Therefore, this approach requires a wind or air flow sensor. In order to stay inside the gas plume or re-discover the plume, in case the robot lost track of it, different approaches were investigated. For example in [85], the sensor readings of multiple gas sensors, spatially distributed on a robot are compared in relation to each other. In this way loosing and detecting the plume can be recognized. The bio-inspired approaches for gas source localization mentioned in the previous section actually imitate mechanisms observed in nature to stay inside the plume or re-discover the plume if it is lost. For example, they follow so called casting trajectories [74] or spiraling trajectories [107] in order to keep track of the plume.

Especially for plume tracking, multiple robots are of an advantage. For example, in [162] and [112] a swarm of robots is controlled so that the robots take an optimal formation with respect to plume tracking. In [69] and [118] a multi-robot system is controlled by particle swarm optimization techniques and in [95] the robots are controlled to arrange themselves along the plume front and to track the plume front over time.

The proposed DARES strategy in this thesis shows implicitly a kind of plume tracking behavior. In fact, based on the DARES strategy robots start to sample the gas distribution in downwind regions and tend to follow the upwind direction towards areas with higher gas concentration. It is remarkable that this behavior is not explicitly programmed or designed. It emerges naturally from the DARES approach and the used gas dispersion model that declares for example downwind regions as informative according to physics of gas dispersion.

2.3.3 Gas Distribution Mapping

Another objective of robotic gas sensing is building gas distribution maps. In contrast to gas source localization where only the localization of the source is of interest, gas distribution mapping approaches aim at computing a map of the gas concentration for the whole environment. Consequently, the gas samples need to be taken more densely and need to cover possibly the whole area. To achieve this goal, a reasonable alternative to an installation of many static sensors are mobile robotic sensors. The high fluctuation and dynamics of gas distributions [86] as well as their patchy structure [113] make gas distribution mapping challenging. Statistical mapping approaches like the Kernel DM+V model [98] are a possible solution. They account for the intermittency of gas distributions by statistic features, e.g. mean and variance. A time dependent version of this approach in [22] can also adapt to changes in the environment. A slightly different but similar approach has been proposed in [113] where the fluctuations in the gas concentration are tackled by low pass filtering. In general, gas concentration maps need to provide a mechanism to interpolate between spatial locations in 2D or 3D [111] where the gas concentration has been measured. It is even possible to map different substances at the same time [30].

It is also of advantage for mapping tasks to incorporate additional knowledge about the environment. For example in [124] information about obstacles is used to build the gas distribution map. In the field of air pollution monitoring building gas distribution or pollution maps is also of significant interest. Here different mobile platforms carrying gas sensors show their advantage such as a Google street car [19], bicycles [71], or small UAVs [182],[17],[41].

The main objective of the DARES strategy proposed in this thesis is to only localize the sources. Building a gas distribution map is not its goal in the first place. However, the developed approach also delivers an estimate of the spatial gas distribution as a side product.

2.3.4 Gas Tomography

In the last decade a new research branch of gas distribution mapping has become popular: gas tomography. It is induced by the property of modern TDLAS sensors that provide integrated gas concentration measurements along a laser beam in contrast to classical in-situ sensors (see also Section 2.1.4). Computer assisted gas tomography aims at building a gas distribution map (or a map of a gas plume) based on multiple integral measurements [177]. Thus, for an accurate gas distribution reconstruction measurement beams from different angles of view are required (ray tracing) and the beams need to cover the whole environment. In case the measurements are carried out by a robot equipped with a TDLAS sensor, accurate information about the robot's position and orientation are required for the reconstruction [31]. In addition, for gas tomography the TDLAS sensors is often combined with a laser rangefinder [20], [131] that allows to compute the length of the beam along which the gas concentration is integrated. Reconstructing a gas distribution map from the measurement beams is more involved compared to in-situ measurements and requires solving a least square optimization problem [177]. The map can be either a two dimensional map generated by ground based robots [31], [20] or even a three dimensional map generated by a flying robotic platform [131]. To generate the gas distribution map efficiently, in [20] the measurement location and the orientation of the measuring robot are optimized. In this way the number of measurements can be minimized while still covering the whole environment.

In contrast to gas tomography, the exploration strategy in this thesis is focused on in-situ gas concentration measurements. However, a future extension to gas tomography would be of interrest. In fact, the used sensor model described in Section 3.4 can be extended to integral measurements for future extensions of the proposed method.

Chapter 3
Gas Dispersion Model for Robotic Exploration

For the Domain-knowledge Assisted Robotic Exploration and Source-localization (DARES) strategy presented in this thesis we need a mathematical description for the dynamic behavior of gas dispersion. This mathematical description is called a “model”. In our context, the purpose of the model is twofold. First based on it we can infer the sources from gas concentration measurements and predict the gas concentration in between measurements locations. Through the use of a model the source localization turns into a parameter estimation problem which will be investigated in chapter 4. Second the mathematical model enables us to quantify uncertainties in the model parameters and informativeness of measurements. The DARES strategy presented in Chapter 6 will be build up on this capability. Besides the two main objectives, the model is further used for forward simulations of gas dispersion in the simulation studies in chapter 7.

In the following we first give a short review of possible models and select the most suitable one based the advection-diffusion PDE. We show how to numerically solve this PDE and present the resulting algebraic system of equations. Besides the dispersion model we introduce a measurement model to characterize the concentration measurements carried out by a multi-robot system. We will comment on the necessity of regularization which will be important for the estimation problem presented in the next chapter. Finally, an illustrating example shows how to apply the develop model.

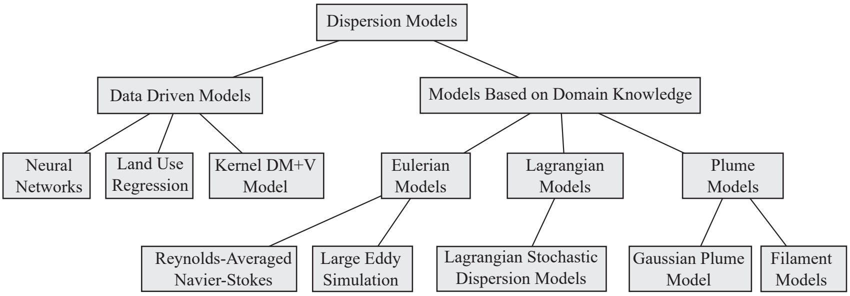


Figure 3.1: Classification of most common gas dispersion models

3.1 Theoretical Models for Gas Dispersion and Mapping

We are interested in modeling the dispersion of a single airborne trace substance. Modeling of dispersion is an active research topic with different applications. Usual applications are in meteorology [57], atmospheric research [94, 165], air quality modeling [44] or in emergency scenarios for evacuation plans [198]. Due to different objectives, different modeling approaches can be found in literature. Here we have a short look at such models that are most appropriate to support robotic exploration. In general, this pertains to small scale dispersion models (<1km) moderately effected by turbulent air flow. Figure 3.1 provides a rough classification of the models surveyed in the following (cf. [143]).

3.1.1 Models Based on Domain Knowledge

Gaussian Plume Model

One of the simplest gas dispersion model is the Gaussian Plume Model. It models airborne transportation of trace substances released by a single point source. The model considers turbulent diffusion and advection caused by wind as the main dispersion mechanism [121]. The model can be analytically derived from the advection-diffusion PDE as a simplified case (see the later section 3.2 for more details on the PDE) under the following main assumptions [165]:

- The material is emitted by a single source with a constant rate.
- The wind velocity is constant and uniform.
- The model only considers a steady state of the plume, i.e. all parameters are independent of time.

- The diffusion in wind direction is much smaller than the advection.

If these assumptions hold, the dispersion plume can be described by 2D Gaussian distributions on planes perpendicular to the downwind direction. The standard deviation of the Gaussian distributions depends on the atmospheric parameters, wind, distance from the source etc. [44]. Even though the model is quite old [147, 167], it builds the foundation of modern, more advanced dispersion models. As an example, the OML (Danish: Operationelle Meteorologiske Luftkvalitetsmodeller) dispersion model is based on a Gaussian Plume model [7]. Also, for gas mapping tasks with robotic mobile sensors the Gaussian plume model was shown to be useful to design a rule-based sampling strategy [92]. Nevertheless, with respect to the robotic source localization task considered in this thesis, the Gaussian Plume model has two disadvantages. First it only considers a steady state of the plume and constant wind conditions, which is a strong limitation for possible applications. Second, classical Gaussian Plume models consider the plume of a single source, while we are looking at scenarios with multiple sources.

Lagrangian and Eulerian Models

Lagrangian and Eulerian modeling approaches are based on the conservation laws (conservation of mass, energy, momentum) [181]. As such, they consider physical knowledge about the dispersion process. The conservation of mass leads to the advection-diffusion PDE - also sometimes referred to as atmospheric transport equation [94]. Both approaches make use of numerical techniques to solve this equation. However, the approaches consider the dispersion process from different points of view. The Eulerian approach makes use of a Cartesian coordinate frame that is static and fixed with respect to the environment. In order to solve the PDE the Eulerian approach discretizes the environment by means of a grid or mesh. In general the Eulerian modeling approaches are computationally very expensive [44]. Most numerical Computational Fluid Dynamics (CFD) simulations, like the later mentioned Large Eddy Simulation (LES) and Reynolds-Averaged Navier-Stokes (RANS), can be classified as Eulerian models. Sometimes even the Gaussian Plume model is classified as a Eulerian model [199], even though it does not require the characteristic discretization.

In contrast to the fixed frame, the Lagrangian approach takes the point of view of the moving dispersive material. The dispersive material or gas is divided into packages - sometimes called puffs - or particles [200, 49]. The Lagrangian approach simulates the movement and trajectories of these packages in quasi-continuous space. Consequently, they suffer less from numerical discretization issues than the Eulerian approach. For calculating the trajectories of the gas packages a Markov process is assumed driven by a PDE

[94]. Due to lower computational complexity the Lagrangian approach was often preferred in the past [171] compared to the Eulerian approach. But even modern dispersion models, e.g. FLEXPART (FLEXible PARTicle dispersion model) [137] or GADEN [122], are based on the Lagrangian model. While the Lagrangian approach is convenient for a forward simulation of gas dispersion with known sources, it is less suitable as an inverse model to estimate a gas source based on concentration measurements. Therefore, in this thesis we will follow the Eulerian approach to get a dispersion model aiding the robotic source localization task.

Computational Fluid Dynamics (CFD) Models

Advanced and sophisticated CFD dispersion simulations make use of the Navier-Stokes equations to account for the physical behavior of airflow [181]. In this way they are able to simulate dispersion based on turbulence. The Reynolds number is a common indicator of turbulent or laminar air flow. With increasing Reynolds numbers turbulence effects get more and more important [57]. CFD simulations numerically approximate of the Navier-Stokes equations based on a discretization of the environment by a 2D/3D grid or mesh. Three different ways to approach the Navier-Stokes equations can be found [181, 114]:

- Direct numerical simulation (DNS) models solve the full Navier-Stokes equations without any further assumptions or simplifications. In order to resolve small scale eddies, a fine discretization is required. In turn the DNS models are computational extremely expensive and only make sense for small Reynolds numbers [202, 114]. On the other side this brute force approach could be considered as quite accurate.
- In Reynolds-Averaged Navier-Stokes (RANS) models a so called Reynolds's decomposition is applied to the variables of the Navier-Stokes equations. As such all variables are described by a mean value (averaged over time) and a fluctuation part [57], where only the mean values are considered to be of interest. By averaging over time the Navier-Stokes equations, new terms appear in the resulting PDEs - the so called Reynolds stress. Due to these new terms the system of equations is underdetermined. In order to solve the RANS, turbulence models are required that provide additional constraints to solve the Reynolds stress terms. The most popular example of such a turbulence model is the $k - \epsilon$ model, which provides two additional PDEs: one for the turbulent kinetic energy k and one for the dissipation of the turbulent kinetic energy ϵ [181]. The RANS models show a moderate computational complexity compared to DNS. However, the model's predictions may be wrong or give inaccurate results in some cases [202].

- The main idea of Large Eddy Simulation (LES) models is to separate the influence of large and small eddies on the air flow. In contrast, RANS models make use of a single turbulence model for all scales of eddies. In order to separate large eddies from small ones, LES models make use of spatial filters on the Navier-Stokes equations. (In contrast RANS averages over time.) Eddies smaller than the cut-off-width of the spatial filters are unresolved in the LES [181]. To account for the interaction of smaller eddies a sub-grid stress term is needed in the filtered Navier-Stokes equations [202]. A common and popular approach is the Smagorinksy-Lilly sub-grid stress model [105, 161]. In general, LES are computationally more expensive compared to RANS. (Of course also depending on the turbulence model of the RANS.) On the other hand LES provides higher resolution in space and time [57] and can be considered more accurate [202].

As a remark, we would like to note that essentially the CFD models describe rather the airflow than the actual gas dispersion. Of course, both phenomena are tightly coupled because of the advection mechanism in the dispersion process. Since an adaptive model-based robotic source localization strategy requires real-time updates of the model and due to the high computational complexity of the mentioned CFD airflow models, we do not consider them in our approach. Nevertheless, with increasing power of modern computers and with improved approximation or simplification techniques CFD models may get suitable for robotic gas mapping or source localization tasks in the future.

3.1.2 Data Driven Models

Land Use Regression (LUR)

Land Use Regression for monitoring air quality and pollution mostly in cities or urban areas was originally proposed in [35]. It is a stochastic model that makes use of concentration measurements and predictor variables [78] (sometimes called covariates). The measurements come from monitoring stations that are spread over the area of interest. Typically, the number of monitoring stations is about 20-100 and they are installed at fixed locations close to roads or other relevant locations [78]. However, in some project also measurements from mobile sensors are used [71, 119, 19]. Since nowadays particulate matter, NO_2 and NO_x are mostly responsible for air pollution in modern cities, LUR models typically focus on these trace substances. LUR provides a method to interpolate the concentration of these substances between the measurement locations. Therefore, additional predictor variables gained from geographic information systems are used. Such variables consider for example the traffic conditions, e.g. by taking into account the distance to the next road and the traffic density on the road. Further, the predictor variables describe land use or land coverage, i.e. distance to industrial areas, housing unit density, etc. Other parameters often considered are population density and topological conditions like altitude over sea level or slope of the terrain. LUR models provide good results for predicting the spatial annual mean concentration [35] to a certain degree. For higher resolutions and hotspot prediction, actual, local measurements are required [119]. At a small scale, short term dispersion model LUR models seem not advisable and thus inappropriate for the considered robotic source localization task in this thesis.

Kernel DM+V

The Kernel DM+V algorithm is popular model for gas distribution mapping in the robotic and MRO community. The model is purely data driven and provides a 2D statistical spatial map of the gas distribution. The model predicts or interpolates the gas distribution in between locations with measurements either collected by a mobile robot or by static sensors. In order to fuse the spatially distributed measurements, a Gaussian kernel function is used to calculate weights for individual measurement locations. The basic idea of this approach was proposed in [97, 98]. The statistical model based on a map for the mean and variance of the gas distribution was introduced in [104]. It is possible to account for wind or air flow in the model. Therefore, in [144] the kernel functions are stretched so that the principle axis of the 2D kernels is aligned with the wind direction. Moreover, the modeling approach provides a confidence map. Loosely speaking, the confidence map shows how many measurements were used to predict the distribution at an unknown location and how strongly they were weighted. The computational complexity of the model depends on the number of measurements and the width of the Gaussian kernel function. In general the complexity is comparably low and suitable for robotic real time applications. Nevertheless, in this thesis we would like to avoid a purely data-driven model, since in the exploration procedure - especial at the beginning - we lack of data. Instead, we would like to incorporate physical knowledge about the gas distribution, which is ignored by purely data driven models.

Neural Networks

Neural Networks are a good regression approach because they can capture strong nonlinearities. Therefore, they are also an option when it comes to gas dispersion modeling. The use of Neural Networks as dispersion models is an on-going research topic that started in the nineties [66, 141]. The approaches can be classified according to the way the network is trained. Most approaches train the network based on measurements from data sets. For example in [138] a Neural Network is used as a dispersion model. It was trained during an experiment where an artificial chemical tracer was released in the Grand Canyon area. The trace substance was measured by several measurement stations. Neural Networks are also used as pollution dispersion models in urban scenarios trained on measurements from air quality monitoring stations [66, 79]. In contrast to training with real concentration measurements, Neural Networks were also trained with artificial data gained from simulations. For example in [50] a Neural Network combined with Cellular Automata is proposed as a dynamic dispersion model for industrial accidents. It is trained based on a data set calculated by means of a RANS simulation. This second approach can be used as a simplification or approximation technique to circumvent the high computational complexity of CFD simulations. Therefore, in the future this modeling approach might be useful for the exploration strategy proposed in this thesis (for further discussion see Section 8.3). On the other hand, Neural Networks trained with real measurement data are less suitable as models for the exploration purpose. At the beginning of an exploration procedure no data are available to train the model. However, one would already like to have a model telling, where to collect data.

3.2 Dispersion Model-based on the Advection-Diffusion PDE

In order to express domain knowledge to assist robotic exploration, for example for gas source localization, we need a model that describes the physical phenomenon of interest sufficiently well while its computational complexity is suitable for online calculations. The latter is required to adaptively incorporate new measurements during the exploration. Unfortunately, the complex physical mechanisms behind gas propagation do not allow a simple and exact mathematical description of the phenomenon. As discussed in the previous section many common gas dispersion model show significant disadvantages in this respect, especially CFD models. Nonetheless, models with low complexity exists that can approximate the dynamics of gas dispersion. Here we make use of an Eulerian approach, where we approximate the complex gas dynamics using a Partial Differential Equation (PDE) with modest complexity.

Within this thesis we use the advection-diffusion PDE to model gas dispersion. The dynamic process is taking place in a bounded and connected spatial domain Ω, which we refer to as the environment. In this thesis we consider the PDE as 2-dimensional, i.e. $\Omega \subset \mathbb{R}^2$, although the true gas dispersion is taking place in 3D. This is a assumption and simplification that is motivated by two facts. First, in our experimental evaluation we consider gas heavier than air and ground based robots that are able to sample only in a plane close to the ground. Second, the 2D approximation keeps the mathematical model feasible for online calculations on our multi-robot system.

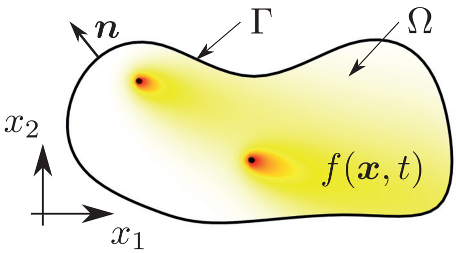


Figure 3.2: This figure illustrates the 2-dimensional domain Ω, where our gas dispersion modeled by a advection-diffusionPDE takes place. The domain Ω is bounded by the border Γ. The color map shows as an example the gas concentration $f(\vec{x}, t)$ driven by two sources located at the peaks of the concentration.

In the following, the continuous function $f(\vec{x}, t) \in \mathbb{R}$ models the gas concentration at point $\vec{x}$ and time t in our exploration environment Ω. Further, the vector valued function $\vec{v}(\vec{x}, t) \in \mathbb{R}^2$ expresses the air flow or wind velocity in the environment. The function $u(\vec{x}, t)$ represents a spatiotemporal source strength at point $\vec{x}$ and time t which drives the evolution of the gas concentration. This function can be considers as the source distribution, since it describes the amount of inflow at point $\vec{x}$ and time t. The advection-diffusion equation (sometimes also convection-diffusion) put these functions into a relation [46]:

$$\frac{\partial f(\vec{x}, t)}{\partial t} - \kappa \nabla^2 f(\vec{x}, t) + \vec{v}(\vec{x}, t)^T \nabla f(\vec{x}, t) = u(\vec{x}, t), \quad \vec{x} \in \Omega,\ t \in \mathbb{R}_+ \tag{3.1}$$

Here, we assume that this velocity field is divergence free, i.e. we assume there are no sources of airflow (e.g. fans, ventilation etc.) inside the environment. The left hand side can be split into three terms:

- The first term is the time derivative of the concentration. Later, we will look at special cases of the PDE. One is the steady state of the dynamic process without any time dependencies that we we gain by setting this time derivative to zero.

- The second term models the diffusion of the concentration. This dispersion mechanism is parametrized by the diffusion coefficient κ in front of the Laplace operator (∇^2). The negative sign causes a flow from regions with high concentration to regions with low concentration. Please note that this term does not only serve to describe the physical molecular diffusion mechanism. Instead, we use it to generally model the isotropic spreading of the gas concentration caused by different effects, including the dispersal and mixing introduced by small local turbulence or eddies (turbulent diffusion).

- The third term describes the propagation of the concentration based on advection. In other words it models the transportation of gas by moving of the air. It is important to note that considering the air flow velocity $\vec{v}(\vec{x}, t)$ and the concentration $f(\vec{x}, t)$ as unknown makes PDE (3.1) a non-linear because of this bi-linear relationship. In absence of air flow or wind in the environment, we will set the advection term to zero. This gives us a linear diffusion PDE. It may be an appropriate approximation in an indoor scenario with very low air flow speeds.

To have a complete and well defined model of the gas dispersion dynamics we further need initial and boundary conditions. As the initial condition we assume:

$$f(\vec{x}, 0) = 0, \vec{x} \in \Omega \tag{3.2}$$

The boundary conditions on the border Γ depend on the structure of the environment. For example the environment maybe bounded by a wall. Alternatively, the environment border can be open so that material can flow off. For the boundary conditions within this thesis we restrict ourselves to two types of homogenous boundary conditions [166]:

$$f(\vec{x}, t) = 0,\ \vec{x} \in \Gamma_d, t \in \mathbb{R}_+, \text{ and} \tag{3.3}$$

$$\frac{\partial f(\vec{x}, t)}{\partial \vec{n}} = 0,\ \vec{x} \in \Gamma_n, t \in \mathbb{R}_+. \tag{3.4}$$

Here we split the border Γ of our environment Ω into two parts: $\Gamma_d \cup \Gamma_n = \Gamma$ Γ_d is the part with the Dirichlet boundary condition (3.3) and Γ_n is the part with the Neumann boundary condition (3.4) where $\vec{n}$ is the outward normal to the boundary. The Dirichlet boundary condition can be physically interpreted as an open bound where material is flowing off so that the concentration at the boundary vanishes (i.e. the concentration at the boundary is zero). In contrast, the Neumann boundary condition can be interpreted as an obstacle or wall where no material is able to flow off (i.e. the concentration gradient perpendicular to the boundary is zero). At this point note that the PDE (3.1) together with the boundary conditions (3.3) and (3.4) is a variational problem, since we are looking for the unknown functions, e.g. the source distribution modeled by $u(\vec{x}, t)$. Such variational problem is hard to handle analytically. Therefore, we are in need of numerical approximation techniques.

3.3 Numerical Approximation

In general solving the PDE (3.1) for the unknown functions $f(x, t), u(x, t), \vec{v}(x, t)$ is analytically impossible. However, numerical methods like the Finite Difference Method (FDM) and the Finite Element Method (FEM) provide approximative solutions for the problem [166, 203, 47]. This section explains how to make use of numerical approximation methods to solve the PDE(3.1) modeling gas dispersion. Loosely speaking, the methods turn the variational problem described by a PDE into the problem of solving a system of algebraic equations. In the following we will derive the system of equations based on a FDM and a FEM approach.

The FDM makes use of a grid based discretization of the environment and sets out by identifying equations that describe the relationships between neighboring cells. Therefore, the results are easier to understand and interpret. A comprehensible system of equations will get more important later on in the thesis when it is shown how to solve the system of equations in a distributed fashion in Section 5.1.

In contrast to an FDM approach, FEM shows more flexibility regarding the spatial discretization schemes. Where in FDM the operators of the PDE are discretized spatially, FEM discretizes the functions of the PDE. Thus, we are no longer restricted to a grid-based discretization and can make use of flexible mesh-based discretization of the environment (compare Figure 3.3 and 3.4). On the downside the resulting system of equations is harder to construe. Both approaches have advantages and disadvantages and we will make use of both during the thesis and choose the most suitable one depending on the case. In the following it is only shown how to apply the numerical approximations to our specific PDE (3.1). For a more detailed introduction to FDM and FEM we refer the interested reader to standard references, e.g. [47],[203],[166].

3.3.1 Finite Difference Method

The FDM approximates differential operators in our PDE (3.1) by finite differences. The calculation of finite differences requires the evaluation of the functions in (3.1) at discrete time stamps and discrete points in space. For the time discretization we consider equidistant time stamps $t = n \cdot \Delta t$ for $n = 0, 1, 2...$ with the sampling period Δt. For the space discretization we divide the two dimensional environment Ω into smaller sub-domains $\Omega_c, c \in \mathcal{C}$ with the cardinality C of the set of cells $\mathcal{C}$. For simplicity, we use quadratic cells Ω_c forming a grid of I rows and J columns, so that $C = J \cdot I$. The width of the cells is denoted as ρ. Moreover, we define the set of grid cells $\mathcal{B} \subset \mathcal{C}$ that belong to the boundary of the environment Ω. Figure 3.3 illustrates this discretization scheme.

The numerical approximation considers the continuous functions $f(x, t)$, $u(x, t)$, $\vec{v}(x, t)$ of the PDE (3.1) as constant within each cell for a time stamp n. Thus, we can describe the concentration field by individual scalar value $f_c[n]$ representing the concentration in cell c at time stamp n. All values $f_c[n]$ are aggregated in a vector $\vec{f}[n] = [f_1[n], f_2[n], ..., f_C[n]]^T$. It is important to note that the row and column indexes i, j map to a unique cell c and the four neigh-

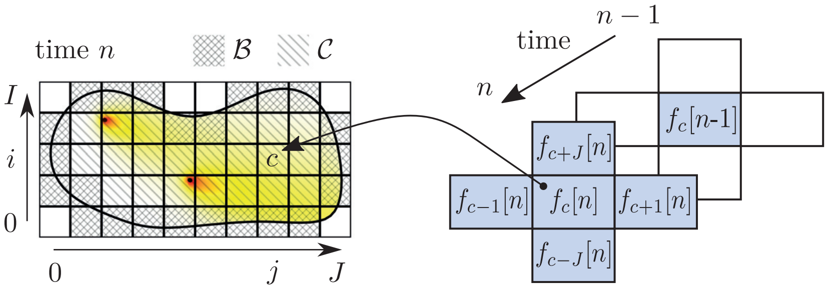


Figure 3.3: The left side illustrates how the continuous domain is discretized by grid cells. The indexes i and j are used to define the location of a cell c in the grid. All cells $c \in \mathcal{C}$ form the domain. The cells $c \in \mathcal{B}$ correspond to the border. Please note the border belongs to the domain, too ($\mathcal{B} \subset \mathcal{C}$). The extract on the right indicates the spatial and temporal neighborhood of a cell c in blue. It shows the four special neighbors as well as a grid cell c at a previous time stamp $n-1$.

boring cells of c are indexed by $c+1$, $c-1$, $c+J$ and $c-J$. Analogously to the concentration, the source distribution vector $\vec{u}[n] = [u_1[n], u_2[n], ..., u_C[n]]^T$ contains the source strength or amount of inflow for each cell. Further, the velocity field is described by two vectors $\vec{v}_1[n], \vec{v}_2[n]$ representing the two spatial components of the velocity in each cell.

Based on the backward Euler method we can approximate the time derivative as:

$$\frac{\partial f(\vec{x},t)}{\partial t} \approx \frac{f_c[n] - f_c[n-1]}{\Delta t}, \tag{3.5}$$

for all cells $c \in \mathcal{C}$ and for the time stamps $n > 0$. Note that we have chosen the implicit backward Euler method because it is more stable compared to a forward Euler method for larger Δt [166]. With a central difference scheme the first and second order spatial derivatives are approximated as:

$$\nabla f(\vec{x},t) \approx \left[\frac{f_{c+1}[n] - f_{c-1}[n]}{\rho}, \frac{f_{c+J}[n] - f_{c-J}[n]}{\rho}\right]^T \tag{3.6}$$

$$\nabla^2 f(\vec{x},t) \approx \frac{-4f_c[n] + f_{c+1}[n] + f_{c-1}[n] + f_{c+J}[n] + f_{c-J}[n]}{\rho^2} \tag{3.7}$$

These difference schemes can be applied for all cells $c \in \mathcal{C}$ that do not belong to the border $\mathcal{B}$ ($c \notin \mathcal{B}$). Further, we introduce an auxiliary vector

$$\vec{s}_c[n] = [f_c[n], f_{c+1}[n], f_{c-1}[n], f_{c+J}[n], f_{c-J}[n], f_c[n-1]]^T \tag{3.8}$$

which collects the concentration value of cell c and the concentration of the neighboring cells $c+1, c-1, c+J, c-J$ as well as the concentration in cell c of the previous time stamp $n-1$. This set of cells is also depicted in Figure 3.3 on the right. Making use of this auxiliary vector and plugging in the finite difference approximations into the PDE (3.1) we obtain the following equation:

$$\vec{h}^T\vec{s}_c[n] - \kappa\vec{d}^T\vec{s}_c[n] + \vec{v}_{1,c}[n]\vec{g}_1^T\vec{s}_c[n] + \vec{v}_{2,c}[n]\vec{g}_2^T\vec{s}_c[n] - u_c[n] = 0, \qquad (3.9)$$
$$c \in \mathcal{C}\setminus\mathcal{B}, \quad n > 0.$$

Note that we move the source term also to the left side. Here, the vectors $\vec{h}$, $\vec{d}$, $\vec{g}_1$, $\vec{g}_2$ are defined so that equation (3.10) reproduces the PDE (3.1) with the discretization scheme of equation (3.5) to (3.7)

$$\begin{aligned}\vec{h} &= \frac{1}{\Delta t}[1,0,0,0,0,-1]^T \qquad (3.10)\\ \vec{d} &= \frac{1}{\rho^2}[-4,1,1,1,1,0]^T\\ \vec{g}_1 &= \frac{1}{\rho}[0,1,-1,0,0,0]^T\\ \vec{g}_2 &= \frac{1}{\rho}[0,0,0,1,-1,0]^T.\end{aligned}$$

The single equation (3.10) holds for a single cell and builds up a system of equations when combined for all cells. We can rewrite this system of equations in a matrix notation. This provides just another formulation that is more suitable for calculations in a computer with state-of-the-art solvers:

$$\frac{1}{\Delta t}\mathbf{H}(\vec{f}[n] - \vec{f}[n-1]) - \kappa\mathbf{D}\vec{f}[n] + \vec{v}_1[n] \circ \mathbf{G}_1\vec{f}[n] + \vec{v}_2[n] \circ \mathbf{G}_2\vec{f}[n] - \vec{u}[n] = 0 \qquad (3.11)$$

where $\circ$ is the Hadamard product, i.e. an element-wise multiplication. The matrices $\mathbf{H}$, $\mathbf{D}$, $\mathbf{G}_1$ and $\mathbf{G}_2$ implicitly contain our discrization scheme (3.5) to (3.7), the auxiliary vector $\vec{s}_c[n]$ and the organization of the grid cells in the vectors $\vec{f}[n]$, $\vec{v}_1[n]$, $\vec{v}_2[n]$ and $\vec{u}[n]$. Unfortunately, it is not easy to explicitly write down the definition of these matrices. Generally they are build - also called assembled - by FDM software tools.

Since we get $|\mathcal{C}\setminus\mathcal{B}|$ equations, the matrices are in $\mathbb{R}^{|\mathcal{C}\setminus\mathcal{B}|x|\mathcal{C}|}$. The vectors $\vec{f}[n]$, $\vec{v}_1[n]$, $\vec{v}_2[n]$ are in $\mathbb{R}^{|\mathcal{C}|}$ and the vector $\vec{u}[n]$ in $\mathbb{R}^{|\mathcal{C}\setminus\mathcal{B}|}$. According to the chosen boundary conditions, the system of equations has to be extended. We obtain one equation for each cell on the boundary. In section 3.2 we split the boundary into a Dirichlet and Neumann bound. Analogously, we divide the set of boundary cells $\mathcal{B}$ into a set $\mathcal{B}_d$ and a set $\mathcal{B}_n$ ($\mathcal{B} = \mathcal{B}_d \cup \mathcal{B}_n$). Here $\mathcal{B}_d$ is the set of cells belonging to the Dirichlet bound and $\mathcal{B}_n$ is the set of cells belonging to the Neumann bound. For the homogenous Dirichlet boundary condition (3.3), we get:

$$f_c[n] = 0, c \in \mathcal{B}_d. \tag{3.12}$$

For a Neumann boundary condition (3.4) we have to approximate the gradient in normal direction with a finite difference scheme. Since the cells are on the border a central difference scheme as in (3.6) is not defined. Instead, a forward or backward difference scheme is required, depending on the location of the boundary, e.g. left or right border. We get:

$$\begin{aligned} f_{c+1}[n] - f_c[n] &= 0 \quad \text{on a left border} \\ f_c[n] - f_{c-1}[n] &= 0 \quad \text{on a right border} \\ f_{c+J}[n] - f_c[n] &= 0 \quad \text{on a lower border} \\ f_c[n] - f_{c-J}[n] &= 0 \quad \text{on a top border} \\ & c \in \mathcal{B}_n. \end{aligned} \tag{3.13}$$

Again all equations of boundary conditions can be arranged in a matrix formulation:

$$\mathbf{B}\vec{f}[n] = 0. \tag{3.14}$$

3.3.2 Finite Element Method

FEMs are another option to obtain a numerical approximation of the PDE. Similar to FDM, FEMs turns our variational problem (3.1) into a system of algebraic equation. In the following we first derive the system of equations for the advection-diffusion PDE as a general case and then discuss special cases in the next section. For simplicity $\dot{f}(\vec{x}, t)$ is used as a short form of $\frac{\partial f(\vec{x},t)}{\partial t}$. Note that during this chapter the notations $[\mathbf{A}]_{i,j}$ and $[\vec{b}]_i$ are used to index an element in row i and column j of a Matrix $\mathbf{A}$ or a vector $\vec{b}$.

The formulation of our PDE (3.1) is called strong form. It has to be fulfilled locally at each point in our domain. FEM, however, makes use of a slightly different formulation of the problem. This formulation is called weak form or sometimes weighted residuals [47]. We obtain this formulation by multiplying the strong form by an arbitrary test function $\phi(\vec{x}, t)$ (sometimes called weighting function) and integrating the product over the whole domain [203]:

$$\begin{aligned} \int_\Omega \phi(\vec{x}, t) \left(\dot{f}(\vec{x}, t) - \kappa\nabla^2 f(\vec{x}, t) + \vec{v}(\vec{x}, t)^T \nabla f(\vec{x}, t)\right) d\vec{x} = \\ = \int_\Omega \phi(\vec{x}, t) u(\vec{x}, t) d\vec{x}. \end{aligned} \tag{3.15}$$

Here, we only consider the two dimensional case. Thus, gradients and the velocity field $\vec{v}(\vec{x}, t)$ have two components. If the weak formulation (3.15) holds for every arbitrary function $\phi(\vec{x}, t)$, then also the strong formulation (3.1) has to hold at every point in the domain [47]. "It is a somewhat surprising fact that often this weak form is more realistic physically than the original differential equation which implied an excessive smoothness of the true solution"[203]. Further, the weak formulation is not just a mathematical trick. It also has a physical meaning. It can be derived via the principle of virtual work. The weak form can be seen as an minimization of a quadratic functional that can be interpreted as a kind of energy [47].

In the weak form (3.15) we can approximate the function space with a corresponding finite dimensional subspace. Therefore, we decompose all continuous functions by a finite number of weighted basis functions $\psi_i(\vec{x})$

$$f(\vec{x}, t) \approx \sum_i f_i(t)\psi_i(\vec{x}), \tag{3.16}$$

$$u(\vec{x}, t) \approx \sum_i u_i(t)\psi_i(\vec{x}), \tag{3.17}$$

with $i = 1, 2, ..., C$ and C being the number of basis functions. In general, the test function $\phi(\vec{x}, t)$ can be chosen arbitrarily. Here we make use of the Galerkin method that uses the same basis functions $\psi_i(\vec{x})$ to construct the test function $\phi(\vec{x}, t)$.

$$\phi(\vec{x}, t) \approx \sum_i \phi_i(t)\psi_i(\vec{x}), \tag{3.18}$$

The basis functions are weighted by the values $f_i(t)$, $u_i(t)$ and $\phi_i(t)$. These values we aggregate in vectors

$$\vec{f}(t) = [f_1(t), f_2(t), ..., f_C(t)]^T, \tag{3.19}$$

$$\vec{u}(t) = [u_1(t), u_2(t), ..., u_C(t)]^T, \tag{3.20}$$

$$\vec{\phi}(t) = [\phi_1(t), \phi_2(t), ..., \phi_C(t)]^T. \tag{3.21}$$

These vectors together with the basis functions approximately represent the continuous functions $f(\vec{x}, t), u(\vec{x}, t), \phi(\vec{x}, t)$. Let us now construct the basis functions $\psi_i(\vec{x})$. To this end we first discretize the spatial environment Ω into small triangles forming a mesh. These triangles are also called elements. The discretization of the environment by such a mesh is illustrated in Figure 3.4 on the left. We make use of Lagrange elements of first order [203]. This means that the basis function are designed in a way so that they linearly interpolate on the triangle the values at triangle's vertexes. We refer to these vertexes as nodes in the following. They build the set $\mathcal{C}$ (cf. grid cells in case of FDM). The weights of the basis functions correspond to the values at the nodes. In total the number of basis function C is the same as the number of nodes ($C = |\mathcal{C}|$). In Figure 3.4 on the right a basis function is plotted as an example. At the corresponding node the basis function is 1 and decays linearly to zero on the neighboring elements. The basis function is zero on all on all other elements. Note again the the weights contained in e.g. $\vec{f}(t)$ are actually the values of the function $\vec{f}(\vec{x},t)$ to be approximated at corresponding node's location.

So far we skipped the wind field $\vec{v}(\vec{x},t)$. Its components are approximated in the same way by a set of C weighted basis functions $\hat{\psi}_i(\vec{x})$:

$$\vec{v}(\vec{x},t) \approx \left[\sum_i v_{1,i}(t)\hat{\psi}_i(\vec{x}), \quad \sum_i v_{2,i}(t)\hat{\psi}_i(\vec{x})\right]^T, \tag{3.22}$$

where we aggregate the weights $v_{1,i}(t)$ and $v_{2,i}(t)$ in vectors

$$\vec{v}_1(t) = [v_{1,1}(t), v_{1,2}(t), ..., v_{1,C}(t)]^T, \tag{3.23}$$

$$\vec{v}_2(t) = [v_{2,1}(t), v_{2,2}(t), ..., v_{2,C}(t)]^T. \tag{3.24}$$

In contrast to the basis functions $\psi_i(\vec{x})$, the basis functions $\hat{\psi}_i(\vec{x})$ are chosen as Lagrange elements of order zero. This implies that the function is 1 within the element and 0 everywhere else (see also Appendix A).

At this point we would like to mention that during the thesis we have only considered basis functions which are defined on single elements and are zero elsewhere. Thus, spectral methods making use of globally defined sinusoids etc. as basis functions [38] were not consider.

In what follows we apply the approximation (3.16)-(3.18) to each term in the weak formulation (3.15) to obtain a system of algebraic equations approximately solving the PDE (3.1).

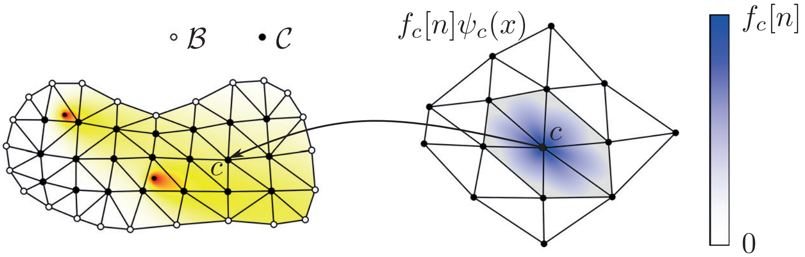


Figure 3.4: The left side shows the spatial discretization of the domain by a mesh of triangles. The nodes indicated with an "o" belong to the border of the domain. Note that these nodes are also in $\mathcal{C}$. The extract on the right illustrates a single basis function $\psi_c(\vec{x})$ associated with the node c and weighted by the value $f_c[n]$. This function is equal to $f_c[n]$ at the node c and decays linearly to zero over the neighboring triangle elements.

Source term:
In the source term of equation (3.15), i.e. the right hand side, we replace $\phi(\vec{x}, t)$ and $u(\vec{x}, t)$ according to (3.18) and (3.17):

$$\begin{aligned}
&\int_\Omega \phi(\vec{x}, t) u(\vec{x}, t) d\vec{x} \approx \qquad (3.25)\\
&\approx \int_\Omega \left(\sum_j [\vec{\phi}]_j(t) \psi_j(\vec{x}) \right) \left(\sum_i [\vec{u}]_i(t) \psi_i(\vec{x}) \right) d\vec{x} = \\
&= \int_\Omega \sum_i \sum_j [\vec{\phi}]_j(t) [\vec{u}]_i(t) \psi_j(\vec{x}) \psi_i(\vec{x}) d\vec{x} \\
&= \sum_i \sum_j [\vec{\phi}]_j(t) [\vec{u}]_i(t) \int_\Omega \psi_j(\vec{x}) \psi_i(\vec{x}) d\vec{x}.
\end{aligned}$$

Further, the summation can be rewritten as a matrix multiplication:

$$\begin{aligned}
&\sum_i \sum_j [\vec{\phi}]_j(t) [\vec{u}]_i(t) \int_\Omega \psi_j(\vec{x}) \psi_i(\vec{x}) d\vec{x} = \qquad (3.26)\\
&= \vec{\phi}(t)^T \tilde{\mathbf{A}} \vec{u}(t); \quad [\tilde{\mathbf{A}}]_{j,i} = \int_\Omega \psi_j(\vec{x}) \psi_i(\vec{x}) d\vec{x}.
\end{aligned}$$

The matrix $\tilde{\mathbf{A}}$ can be assembled by any FEM software tool. Further note that since the basis function $\psi_i(\vec{x})$ are just piecewise defined function on elements, the integral over the whole domain Ω becomes a sum of integrals over all elements, which can be calculated analytically.

Diffusion term:
For the diffusion term in equation (3.15) we can do integration by parts and apply the Green's theorem:

$$\begin{aligned}
&-\kappa \int_\Omega \phi(\vec{x}, t) \nabla^2 f(\vec{x}, t) d\vec{x} = \qquad (3.27)\\
&= \kappa \int_\Omega \nabla \phi(\vec{x}, t)^T \nabla f(\vec{x}, t) d\vec{x} - \kappa \int_\Gamma \phi(\vec{x}, t) \frac{\partial f(\vec{x}, t)}{\partial \vec{n}} d\vec{x} = \\
&= \kappa \int_\Omega \nabla \phi(\vec{x}, t)^T \nabla f(\vec{x}, t) d\vec{x}.
\end{aligned}$$

To achieve the last line we have to demand that either $\phi(\vec{x}, t) = 0$ or $\frac{\partial f(\vec{x}, t)}{\partial \vec{n}} = 0$ on the boundary Γ. The second condition is fulfilled on Neumann boundaries Γ_n naturally. Therefore, the Neumann boundary condition is implicitly considered solving the weak formulation. The Dirichlet boundary Γ_d will be considered separately later on. With the approximations (3.18) and (3.16) we get

$$\kappa\int_\Omega \nabla\phi(\vec{x},t)^T\nabla f(\vec{x},t)d\vec{x} = \tag{3.28}$$
$$= \kappa\int_\Omega \left(\sum_j [\vec{\phi}]_j(t)\nabla\psi_j(\vec{x})^T\right)\left(\sum_i [\vec{f}]_i(t)\nabla\psi_i(\vec{x})\right)d\vec{x} =$$
$$= \kappa\int_\Omega \sum_j\sum_i [\vec{\phi}]_j(t)[\vec{f}]_i(t)\nabla\psi_j(\vec{x})^T\nabla\psi_i(\vec{x})d\vec{x}.$$

As can be seen, because of the partial integration of the weak formulation it is sufficient that the basis functions are continuously differentiable. Again, the summation can be rewritten as a matrix multiplication:

$$\kappa\int_\Omega \sum_j\sum_i [\vec{\phi}]_j(t)[\vec{f}]_i(t)\nabla\psi_j(\vec{x})^T\nabla\psi_i(\vec{x})d\vec{x} = \tag{3.29}$$
$$= \kappa\vec{\phi}(t)^T\tilde{\mathbf{D}}\vec{f}(t)$$
$$[\tilde{\mathbf{D}}]_{j,i} = \int_\Omega \nabla\psi_j(\vec{x})^T\nabla\psi_i(\vec{x})d\vec{x},$$

where the matrix $\tilde{\mathbf{D}}$ can be assembled by any FEM software tool.

Advection term:
For the advection term of equation (3.15) we make use of the approximation (3.16) and (3.22).

$$\int_\Omega \phi(\vec{x},t)\vec{v}(\vec{x},t)^T\nabla f(\vec{x},t)d\vec{x} \approx \tag{3.30}$$
$$\approx \int_\Omega \left(\sum_j [\vec{\phi}]_j(t)\psi_j(\vec{x})\right)\left(\sum_k [\vec{v}_1]_k(t)\hat{\psi}_k(\vec{x})\right)\left(\sum_i [\vec{f}]_i(t)[\nabla\psi_i(\vec{x})]_1\right)d\vec{x}+$$
$$+ \int_\Omega \left(\sum_j [\vec{\phi}]_j(t)\psi_j(\vec{x})\right)\left(\sum_k [\vec{v}_2]_k(t)\hat{\psi}_k(\vec{x})\right)\left(\sum_i [\vec{f}]_i(t)[\nabla\psi_i(\vec{x})]_2\right)d\vec{x} =$$
$$= \vec{\phi}(t)^T\vec{v}_1(t)\circ\tilde{\mathbf{G}}_1\vec{f}(t) + \vec{\phi}(t)^T\vec{v}_2(t)\circ\tilde{\mathbf{G}}_2\vec{f}(t)$$

with

$$[\tilde{\mathbf{G}}_1]_{j,i} = \int_\Omega \psi_j(\vec{x})[\nabla\psi_i(\vec{x})]_1 d\vec{x}, \quad [\tilde{\mathbf{G}}_2]_{j,i} = \int_\Omega \psi_j(\vec{x})[\nabla\psi_i(\vec{x})]_2 d\vec{x}. \tag{3.31}$$

Here, in order to rewrite the summation as a matrix multiplication, we make use of the Hadamard product, i.e. an element-wise multiplication. Since the derivation in equation (3.31) is more involved, a more detailed derivation with intermediate steps is given in Appendix A. Again the matrices $\tilde{\mathbf{G}}_1$,$\tilde{\mathbf{G}}_2$ can be obtained from standard FEM software tools.

Time derivative term:
Finally, the term in equation (3.15) containing the time derivative of $f(\vec{x}, t)$ is approximated analogously to the source term.

$$\begin{aligned}&\int_\Omega \phi(\vec{x}, t)\dot{f}(\vec{x}, t)dx \approx \\ &\approx \int_\Omega \left(\sum_j [\vec{\phi}]_j(t)\psi_j(\vec{x})\right)\left(\sum_i [\dot{\vec{f}}]_i(t)\psi_i(\vec{x})\right)d\vec{x} = \\ &= \vec{\phi}(t)^T\tilde{\mathbf{A}}\dot{\vec{f}}(t); \quad \tilde{\mathbf{A}}[j,i] = \int_\Omega \psi_j(\vec{x})^T\psi_i(\vec{x})d\vec{x}\end{aligned} \tag{3.32}$$

Putting all results together in a system of algebraic equations:

$$\tilde{\mathbf{A}}\dot{\vec{f}}(t) + \kappa\tilde{\mathbf{D}}\vec{f}(t) + \vec{v}_1(t) \circ \tilde{\mathbf{G}}_1\vec{f}(t) + \vec{v}_2(t) \circ \tilde{\mathbf{G}}_2\vec{f}(t) - \tilde{\mathbf{A}}\vec{u}(t) = 0 \tag{3.33}$$

We further simplify the equations by multiplying with $\tilde{\mathbf{A}}^{-1}$ which results in a new system of equations. As can be seen from the definition of $\tilde{\mathbf{A}}$ in equation (3.33), it symmetric an positive definite, since the basis functions are always positive. Therefore, it's inverse exists. Note that we also switch the sign of the diffusion term so that the algebraic system is congruent with the original PDE (3.1).

$$\begin{gathered}\dot{\vec{f}}(t) - \kappa\mathbf{D}\vec{f}(t) + \vec{v}_1(t) \circ \mathbf{G}_1\vec{f}(t) + \vec{v}_2(t) \circ \mathbf{G}_2\vec{f}(t) - \vec{u}(t) = 0 \\ \mathbf{D} = \tilde{\mathbf{A}}^{-1}\tilde{\mathbf{D}}; \quad \mathbf{G}_1 = \tilde{\mathbf{A}}^{-1}\tilde{\mathbf{G}}_1; \quad \mathbf{G}_2 = \tilde{\mathbf{A}}^{-1}\tilde{\mathbf{G}}_2\end{gathered} \tag{3.34}$$

Solving this system of equations provides a numerical approximation of the original variational problem (3.1). Instead of the spatial functions $f(\vec{x}, t)$, $u(\vec{x}, t)$, $\vec{v}(\vec{x}, t)$ we are looking for the vectors $\vec{f}(t)$, $\vec{u}(t)$, $\vec{v}_1(t)$, $\vec{v}_2(t)$. However, these vectors are still time dependent functions. A numerical implementation requires a further discretization of these time continuous functions. Therefore, we make use of the backward Euler method to numerically approximate the time derivative. Instead of vector-valued functions e.g. $\vec{f}(t)$ with respect to time we get a vector $\vec{f}[n]$ at each discrete time stamp $t = n \cdot \Delta t$ for $n = 0, 1, 2, ...$ and the sampling period Δt. Note that in this context the index n does not indicate a single element of a vector but represents the whole vector at a certain time stamp n. We obtain:

$$\frac{1}{\Delta t}\mathbf{H}(\vec{f}[n] - \vec{f}[n-1]) - \kappa\mathbf{D}\vec{f}[n] + \vec{v}_1[n] \circ \mathbf{G}_1\vec{f}[n] + \vec{v}_2[n] \circ \mathbf{G}_2\vec{f}[n] - \vec{u}[n] = 0. \tag{3.35}$$

Essentially, here the matrix $\mathbf{H}$ is the identity matrix. We introduce this matrix to have the same structure as in equation (3.11). Note that in contrast to equation (3.11) all matrices in equation (3.35) are in $\mathbb{R}^{|\mathcal{C}| \times |\mathcal{C}|}$ and the vectors are in $\mathbb{R}^{|\mathcal{C}|}$. When deriving this equation, we made use of the Green's Theorem in equation (3.27). There the Neumann boundary condition (3.4) was implicitly used so that it is already considered in equation (3.35). This is the reason why the number of equations, i.e. number of matrix rows, is higher in case of FEM compared to FDM.

However, the homogenous Dirichlet boundary condition (3.3) has to be considered separately, too. It can be simply superposed to equation (3.35). Therefore, we define a subset of our nodes $\mathcal{B}_d \subset \mathcal{C}$ which belong to the Dirichlet border of Γ_d. For these nodes we get the additional equations:

$$[\vec{f}]_c[n] = 0, c \in \mathcal{B}_d. \tag{3.36}$$

This can be also written in a matrix form:

$$\mathbf{B}\vec{f}[n] = 0, \tag{3.37}$$

where $\mathbf{B}$ selects the node values contained in $\vec{f}[n]$ that are on the Dirichlet border.

3.3.3 Results

Both approximation methods turn the general gas dispersion PDE (3.1) into a system of algebraic equations (equation (3.35) for FEM and equation (3.11) for FDM plus boundary conditions). Essentially, in the matrix form they have exactly the same structure. However, they are not exactly the same since the content of the matrices is different and also the content of the vectors has a different interpretation. Nevertheless, the same numeric solvers can be applied to both systems of equations. In the following, we collect the results from the last two sections. Further, we will show the different regimes of the PDE and discuss them as special cases. In the previous sections, the systems of equations where always set to zero, e.g. equation (3.35) or equation (3.11) or the boundary conditions (3.14), (3.37). Now, we are going to introduce a residual vector $\vec{r}[n]$, instead of setting the equations to zero. This is a first perpetration toward the probabilistic formulation in the next chapter. While in the current chapter we demand the residuals to be zero, we will relax this constraint in the next chapter.

Dynamic Advection-Diffusion:
We derived the numerical approximation starting from the most general case considered in this thesis, i.e. advection-diffusion based dynamic dispersion:

$$\frac{1}{\Delta t}\mathbf{H}(\vec{f}[n]-\vec{f}[n-1])-\kappa\mathbf{D}\vec{f}[n]+\vec{v}_1[n]\circ\mathbf{G}_1\vec{f}[n]+\vec{v}_2[n]\circ\mathbf{G}_2\vec{f}[n]-\vec{u}[n]=\vec{r}[n], \tag{3.38}$$

As long as the concentration $\vec{f}[n]$ and the wind velocity field $\vec{v}_1[n],\vec{v}_2[n]$ are treated as unknowns, equation (3.38) presents a non-linear problem, because of the bi-linearity in the advection term. If we consider the wind velocity field $\vec{v}_1[n],\vec{v}_2[n]$ as deterministically known, the problem turns into a linear system of equations.

Static Advection-Diffusion:
As a special case, we will consider later on the steady state of the advection-diffusion gas dispersion process. For the static case time derivatives in the PDE vanish. In other words the change between two successive time stamps is zero, which results in:

$$-\kappa\mathbf{D}\vec{f}[n]+\vec{v}_1[n]\circ\mathbf{G}_1\vec{f}[n]+\vec{v}_2[n]\circ\mathbf{G}_2\vec{f}[n]-\vec{u}[n]=\vec{r}[n] \tag{3.39}$$

Again depending whether the wind velocity field is known or not, we have to solve a linear or a non-linear system of equations.

Dynamic Diffusion:
Another special case considered in this thesis is gas dispersion without an advection mechanism, e.g. an indoor scenario where airflow can be neglected. Setting the velocity field to zero, simplifies the problem to:

$$\frac{1}{\Delta t}\mathbf{H}(\vec{f}[n]-\vec{f}[n-1])-\kappa\mathbf{D}\vec{f}[n]-\vec{u}[n]=\vec{r}[n]. \tag{3.40}$$

In this case the problem is linear.

Static Diffusion:
The problem gets even simpler if we consider the steady state without advection. By setting the time derivatives to zero we get the linear system of equations:

$$-\kappa\mathbf{D}\vec{f}[n]-\vec{u}[n]=\vec{r}[n]. \tag{3.41}$$

Boundary Conditions:
The boundary conditions can be obtained by equation (3.37) in case of FEM or by equation (3.14) in case of FDM.

$$\mathbf{B}\vec{f}[n]=\vec{b}. \tag{3.42}$$

Here we also introduced a residual vector $\vec{b}$. Note that these boundary conditions have to be combined with any of the previously mentions cases. Otherwise, the system of equations will be ill-posed.

3.4 Multi-Robot Measurement Model

For inspection robots that explore an environment, the central goal is to estimate the unknown vectors $\vec{f}[n]$, $\vec{v}_1[n], \vec{v}_2[n]$ and $\vec{u}[n]$ based on the collected measurements. We consider L ground based robots equipped with gas sensors to measure the gas concentration $\vec{f}[n]$. The robots are capable of driving to requested 2D locations in the environment and are able to carry out measurements there. The focus of this thesis is on in-situ gas sensing. Thus, we will assume that the width of a sensor point spread function (sensor footprint) is essentially infinitesimal small. This assumption is justified considering a relatively rough discretization of the environment compared to the sensor's size. Further, gas sensors often have a low-pass time characteristic due to physical constraints of the measurement principle. However, we consider the time of the impulse response of the sensor as small compared to the time discretization Δt of the numerical approximation. This implies that the actual process measurement $y_l[n]$ of the robot l only depends on the current concentration distribution $\vec{f}[n]$ as:

$$y_l[n] = \vec{m}_l^T[n]\vec{f}[n] + \xi_l[n], \quad l = 1, 2, ...L. \tag{3.43}$$

Here, the vector $\vec{m}_l[n]$ selects a single element from the concentration distribution $\vec{f}[n]$. This vector depends on the position of the robot and since the robots is moving is time dependent. In other words, it is a zero vector except for one element that is 1 at an index that corresponds to the discrete spatial cell or mesh node measured by the robot l at time n.

The measurements are disturbed by additive noise $\xi_l[n]$. We assume our noise to be a spatially and temporally white, normally distributed process with zero mean and precision (a reciprocal of the variance) τ_m, i.e., $\xi_l[n] \sim N(0, \tau_m^{-1})$.

In a realistic and efficient implementation of an exploration strategy, it is not practical to require synchronization of the robots so that we get a measurement of each robot at each time stamp. For example a robot may have to move a longer distance to the next measurement location compared to others. Thus, a synchronized measurements procedure would require all robots to wait for it. Instead, we design a system where we can plug in measurements of robots whenever they become available. This implies that measurements of a robot l are not necessarily available for all time stamps n.

In the following, the individual measurements $y_l[n]$ are aggregated in a measurement vector $\vec{y}[n]$. Based on (3.43) we can write:

$$\vec{y}[n] = \mathbf{M}[n]\vec{f}[n] + \vec{\xi}[n], \tag{3.44}$$

where $\mathbf{M} = [\vec{m}_0[n], ..., \vec{m}_L[n]]^T$. In case a robot l does not provide a measurement at time n, the corresponding row $\vec{m}_l[n]$ in $\mathbf{M}$ is missing. As such, the dimensions of $\vec{y}[n]$ is not the same for all time stamps n, since the number of measurements is varying.

3.5 Sparse Regularization

The numerical models descibed above can be used in different ways depending on which values are known or unknown. If the source distribution represented by $\vec{u}[n]$ and the air flow velocity represented by $\vec{v}_1[n], \vec{v}_2[n]$ are known, we can simulate the gas concentration $\vec{f}[n]$. We refer to this as the forward problem. In combination with sufficient boundary conditions this problem is well posed. In other words the solution is unambiguous. While we can use the forward problem to build a simulator for gas dispersion, for exploration and gas source localization we have to solve the inverse problem. In the inverse problem the source distribution $\vec{u}[n]$ is unknown as well as the gas concentration $\vec{f}[n]$. However, we have additional information about the concentration based on collected measurements by a gas sensor. Therefore, the system of equations can be extended by the measurement model (3.43). Nevertheless, even by incorporating measurements, the inverse problem is ill posed, especially in the early phase of the exploration procedure, where no or only few measurements are available. This fact requires a regularization of the problem. At this point we can make use of further a-priori knowledge about the observed gas dispersion process. In the mathematical model (3.1) the source distribution is modeled by the continuous function $u(\vec{x}, t)$ and numerically approximated by the vector $\vec{u}[n]$. This modeling approach shows maximal flexibility and has no restrictions regarding the number of sources. However, in a realistic source localization scenario we would expect $u(\vec{x}, t)$ to be a composition of a few single peaks rather than a continuous function. In other words, even though we do not know the exact number of sources in a realistic scenario we can assume a few sparsely distributed sources. This a-priori knowledge can be exploited in the regularization of the problem. Since the solution of the inverse problem is ambiguous, we choose a regularization that provides us the solution with the sparsest source distribution.

After the numerical approximation the vector $\vec{u}[n]$ represents the source distribution. More precisely, each element $u_c[n]$ of $\vec{u}[n]$ represents the source strength in the grid cell c for an FDM approach, or at a mesh node c in case of a FEM approach. In short, a sparse source distribution corresponds to a parse vector $\vec{u}[n]$. Mathematically the regularized numerical problem can be written as:

$$\min_{\vec{u}[n]} \|\vec{u}[n]\|_{l_0}, \tag{3.45}$$

subject to the equations (3.38)-(3.41) (depending on the dispersion process) and the boundary conditions (3.14) or (3.37). Here the l_0 pseudo norm represents the number of non-zero elements in the vector $\vec{u}[n]$. This kind of regularization is common in compressed sensing [59, 37, 53]. Compressed sensing techniques try to reconstruct signals from fewer samples than the Shannon sampling theorem would require assuming that the signal could be expressed by a sparse vector in some domain [67]. In general solving the l_0 optimization is NP hard. However, if the optimization is subject to a linear system that fulfills the restricted isometry property, the l_0 norm can be replaced by an l_1 norm [42]. In this case the optimization will often be still able to reconstruct sparse signals exactly.

The techniques could be applied for compression or de-noising of signals but it is also quite often applied to the regularization problem like considered here. For example in connection with PDEs, sparsity has been used in [116], where l_1 regularization was used for estimating activation time of a known number of sources. In contrast to temporal sparsity, as in [116], an l_1-regularization was used in [91] for identification of a spatially-sparse right-hand side of a PDE using FEM discretization and measurements from non-mobile sensors. Similarly, [77] covers the problem of sparse input for PDEs but in the context of optimal actuator placement.

The Least Absolute Selection and Shrinkage Operator (LASSO) is one option to tackle the l_1 regularization [173]. LASSO can be considered as an l_1 constrained least square problem [134]. For a linear system, e.g. (3.41) $\mathbf{D}\vec{f} = \vec{u}$, it can be writen as [134]:

$$\min_{\vec{u}[n]} \left\| \mathbf{D}\vec{f} - \vec{u} \right\|_{l_2} + \alpha \left\| \vec{u} \right\|_{l_1}, \tag{3.46}$$

During this thesis we will make use of the state-of-the-art LASSO as a benchmark algorithms for sparse regularization. However, in the next chapter we will transform the estimation problem to a probabilistic framework, where we can apply more powerful and more efficient techniques to archive a sparse regularization.

3.6 Illustrating Example

Let's apply the methods and tools presented so far to an example. For a simple example we consider the case of a static purely diffusion based gas dispersion. We make use of FEM as a numerical approximation. The environment and the discretized mesh are illustrated in Figure 3.5. The white circles indicate the mesh nodes belonging to the border of the environment. This case corresponds to equation (3.41), where we set the diffusion coefficient $\kappa = 1$. Thus we get the linear system of equations:

$$\mathbf{D}\vec{f} + \vec{u} = 0. \tag{3.47}$$

Here we only look at a single time stamp and skip an explicit dependency of variables on the time index n. Further, we get additional constraints from our boundary conditions. We consider here an Dirichlet bound assuming that the concentration at the border of the environment is zero:

$$\mathbf{B}\vec{f} = 0. \tag{3.48}$$

In our example the mesh consists of 500 nodes, where 85 belong to the border. Hence, the vectors $\vec{f}$ and $\vec{u}$ are in $\mathbb{R}^{500}$. Further: $\mathbf{D} \in \mathbb{R}^{500x500}$ and $\mathbf{B} \in \mathbb{R}^{85x500}$.

Let us first simulate the gas concentration $\vec{f}$ given a source distribution $\vec{u}$, i.e. the forward model. For our example we placed two sources: one at location $x_1 \approx 46, x_2 \approx 46$ with strength 0.8 and one at location $x_1 \approx 65, x_2 \approx 101$ with strength 1.0. Therefore, our the source distribution vector $\vec{u}$ contains only zeros except for one element with 0.8 and one with 1.0. This source distribution is shown in Figure 3.5(a). We can combine equations (3.47) and (3.48):

$$\begin{bmatrix} \mathbf{D} \\ \mathbf{B} \end{bmatrix} \vec{f} + \begin{bmatrix} \vec{u} \\ \vec{0} \end{bmatrix} = 0 \tag{3.49}$$

$$\mathbf{S}\vec{f} + \vec{q} = 0$$

We can calculate the concentration by

$$\vec{f} = \mathbf{S}^{+}\vec{q}, \tag{3.50}$$

where $\mathbf{S}^{+}$ denotes the pseudo inverse of $\mathbf{S}$. Here we introduce the auxiliary vector $\vec{q} = [\vec{u}^{T} \quad \vec{0}^{T}]^{T}$ and the system matrix $\mathbf{S} = [\mathbf{D}^{T} \quad \mathbf{B}^{T}]^{T}$.

The result of the concentration distribution is shown in Figure 3.5(b).

Now let us also have a look at the inverse case needed for the exploration. For this case we would like to estimate the source distribution $\hat{\vec{u}}$ given measurements $\vec{y}$. In our example we generate the measurements from our simulated gas concentration $\vec{f}$.

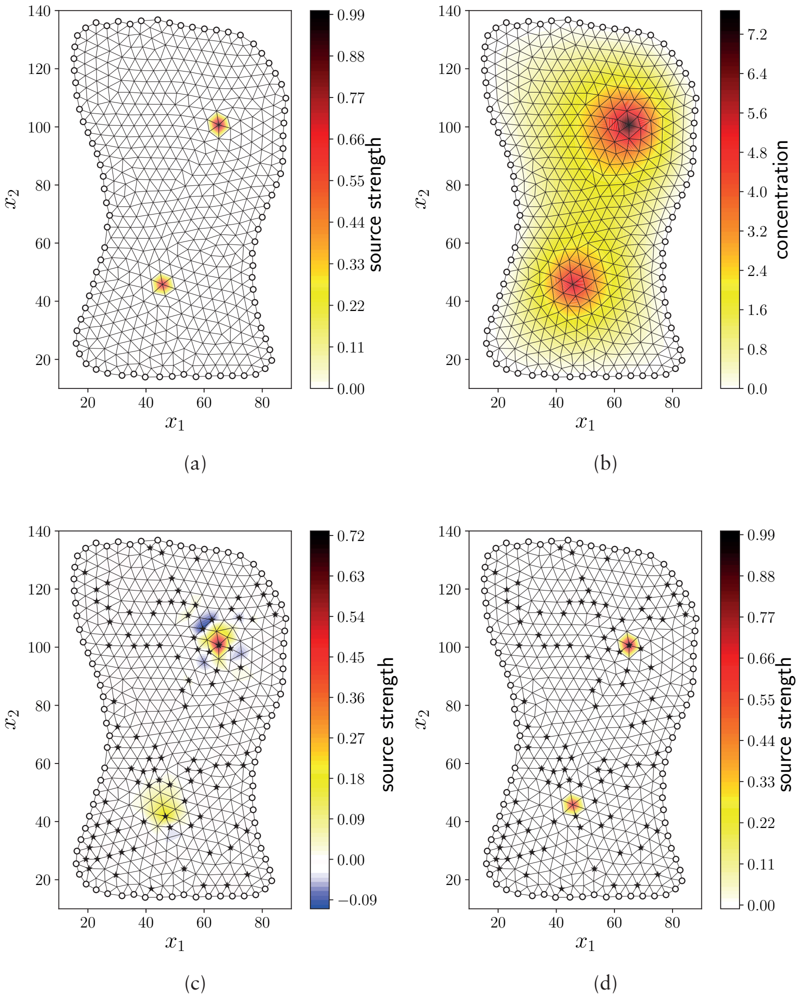
$$\vec{y} = \mathbf{M}\vec{f} + \vec{\xi} \tag{3.51}$$

We pick 100 random locations in our environment as measurements. These locations are reflected in the matrix $\mathbf{M}$ which maps all concentration values $\vec{f}$ to the 100 measurements aggregated in $\vec{y} \in \mathbb{R}^{100}$. The additive noise $\vec{\xi}$ is sampled from a normal distribution center at zero with variance 10^{-4}. By plugging in equation (3.50) into equation (3.51) we get:

$$\vec{y} = \mathbf{M}\mathbf{S}^{+} \begin{bmatrix} \vec{u} \\ \vec{0} \end{bmatrix}. \tag{3.52}$$

Unfortunately the matrix $\mathbf{MS}^{+}$ is underdetermined. (It has only rank 100.) So it is not possible to estimate the source distribution $\vec{u}$ without a regularization. Let's try to treat it as a regularized least square problem:

$$\hat{\vec{u}} = \arg\min_{\vec{u}} \left\| \vec{y} - \mathbf{M}\mathbf{S}^{+} \begin{bmatrix} \vec{u} \\ \vec{0} \end{bmatrix} \right\|_{l_2} + \|\alpha_{u}\vec{u}\|_{l_2} \tag{3.53}$$



(a)

(b)

(c)

(d)

Figure 3.5: The Figure illustrates the source estimation problem. In (a) the ground truth source distribution is shown. This was design arbitrarily. In (b) the gas concentration distribution is plotted. This distribution is calculated based on the source distribution in (a). The plots in (c) and (d) depict reconstructions of the original source distribution (a) based on measurements at locations indicated with the black stars. In (c) a least square regularization is used and LASSO in (d).

We found that for our example a regularization parameter $\alpha_{ll} = 10^{-3}$ achieves the best result. For implementation the least square algorithm of the numpy library [2] was used. The estimate of the source distribution $\hat{\vec{u}}$ is shown in Figure 3.5(c). From the visual impression the estimate is not so bad, however there is a clear discrepancy compared to the original source distribution shown in Figure 3.5(a). Moreover the result is not sparse as requested.

Let's consider an alternative regularization. As discussed in the previous section, we can formulate the inverse problem as a LASSO problem:

$$\hat{\vec{u}} = \arg\min_{\vec{u}} \left\| \vec{y} - \mathbf{M}\mathbf{S}^{+} \begin{bmatrix} \vec{u} \\ \vec{0} \end{bmatrix} \right\|_{l_2} + \|\alpha_{lasso}\vec{u}\|_{l_1} \tag{3.54}$$

Again we found a regularization parameter $\alpha_{lasso} = 10^{-3}$ as a good choice. In the implementation we used the scikit-learn [4] library. The resulting source distribution $\hat{\vec{u}}$ is plotted in Figure 3.5(d). This time the result corresponds very well to the original source distribution shown in Figure 3.5(a). The estimate clearly shows the two original source peaks at the right location an with the right strength. Further, the estimate does not contain any additional, artificial artifacts as in case of the least square regularization in Figure 3.5(c).

This example illustrates that it is possible to estimate the source distribution based on the underdetermined model and only a few measurements. Further, it shows the importance of an appropriate regularization to achieve sparse and good results. Let us remark again that we were able to estimate the 415 unknown elements of the source distribution vector $\vec{u}$ by only 100 measurements.

3.7 Summary

In this chapter we found the advection-diffusion PDE to be a suitable mathematical model for gas dispersion. We explained how to approximate the functional problem by numerical methods in order to translate the advection- diffusion PDE to a system of algebraic equations. Further, we introduced a measurement model characterizing the gas concentration measurements of our multi-robot system. Last but not least we discussed that estimating sources based on the dispersion model and measurements requires regularization. A small example illustrated how to apply the tools introduced in this chapter in a deterministic setup to estimate a source distribution based on measurements. In the next chapter we are going to look at the source estimation problem in a probabilistic framework.

Chapter 4
Probabilistic Problem Formulation

In the previous chapter we introduced a mathematical model for the gas dispersion process based on the advection-diffusion PDE. In what follows, we are going to translate this deterministic model into a probabilistic framework. In the probabilistic framework all variables, vectors and most parameters are treated as random variables. This allows us to apply Bayesian inference to estimate the gas sources based on concentration measurements. Further, the probabilistic framework will be used on Chapter 6 to design the Domain-knowledge Assisted Robotic Exploration and Source-localization (DARES) strategy for a multi-robot system.

4.1 Bayesian Inference for Source Estimation

Mathematically, the source localization problem means to find the source distribution based on observed concentration measurements. This is a classical estimation problem in the field of statistics. The gas dispersion can be seen as a stochastic process.

Let us consider the process described by equation (3.38) that we obtain by a numerical approximation of the advection-diffusion PDE (3.1) in the previous chapter. The gas concentration $\vec{f}[n]$ at a time stamp n depends on the gas concentration of the previous time stamp $\vec{f}[n-1]$. Moreover, it depends on the source distribution $\vec{u}[n]$ and air flow $\vec{v}_1[n]$,$\vec{v}_2[n]$. The Bayesian network in Figure 4.1 illustrates this relation. Further, our observation $\vec{y}[n]$ depend on the current gas concentration $\vec{f}[n]$, according to the measurement model presented in section 3.4. As can be seen from the Bayesian network in Figure 4.1, the source distribution is not directly measurable. The network can be identified as a Hidden Markov model, with the hidden variable $\vec{u}[n]$, since the all variables at time stamp n only depend on the previous concentration $\vec{f}[n-1]$. In contrast to the deterministic point of view of the previous chapter, we now treat

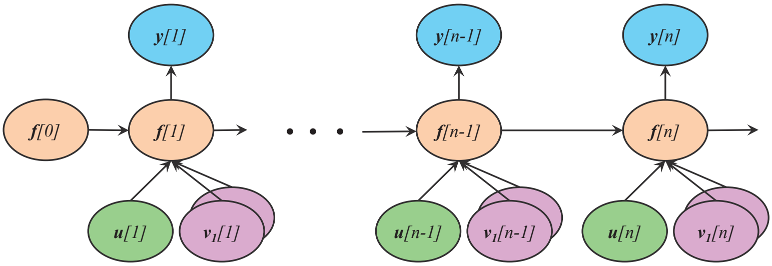


Figure 4.1: This figure shows the Bayesian network of the gas dispersion process for time stamp n. It can be identified as a Hidden Markov model, where the source distribution is the hidden variable.

all variables and vectors in the Bayesian network shown Figure 4.1 as random variables. As such, they are described by PDFs.

We propose as Bayesian approach. Our main objective is to find a PDF describing the source distribution. To this end we first aim at the joint PDF $p(\vec{f}[n], \vec{u}[n], \vec{v}_1[n], \vec{v}_2[n], \vec{f}[n-1] | \vec{y}[0], ..., \vec{y}[n])$ of source distribution, concentration and - in the most general case - air flow. Based on the joint PDF it is then easy to obtain the source distribution PDF by marginalizing over all other variables. In the same way we can get the concentration PDF out of the joint PDF. Note that our may focus is to estimate the source distribution. However, we will see that we get the gas concentration nearly for free, as a side product.

The ingredients of our proposed Bayesian approach are (i) the likelihood given by the measurement model, (ii) the gas dispersion model describing the transition from time stamp $n-1$ to n and (iii) prior distributions. The prior distributions can be selected freely, and are designed by us in a way to encode a-priori known information. Using the Bayes theorem and Figure 4.1, the posterior joint PDF for a single time stamp n is given by

$$\begin{aligned}
& p(\vec{f}[n], \vec{u}[n], \vec{v}_1[n], \vec{v}_2[n], \vec{f}[n-1] | \vec{y}[0], ..., \vec{y}[n]) \propto && (4.1)\\
\text{(i)}\quad & p(\vec{y}[n] | \vec{f}[n]) \\
\text{(ii)}\quad & p(\vec{f}[n] | \vec{u}[n], \vec{v}_1[n], \vec{v}_2[n], \vec{f}[n-1]) \\
\text{(iii)}\quad & p(\vec{v}_1[n], \vec{v}_2[n]) \\
\text{(iv)}\quad & p(\vec{u}[n]) \\
\text{(v)}\quad & p(\vec{f}[n-1] | \vec{y}[0], ..., \vec{y}[n-1]),
\end{aligned}$$

according to the Markov assumption. So the posterior is a factorized function of five terms that can be interpreted as follows:

(i) This term is the likelyhood arising from the measurement model (3.43). It will be further discussed in the next Section 4.2.

(ii) This term is the transition of the concentration distribution between two time stamps. This is the probabilistic equivalent of the gas dispersion model presented in Section 3.3.3. It will be further discussed in the next Section 4.2, too.

(iii) This term is the prior distribution of the air flow or wind. Through this term we can incorporate a-priori information about the air flow or wind velocity. We will have a closer look at this prior in Section 4.3.

(iv) This term is the prior PDF of the source distribution. In this prior we will encode the a-priori assumption that our sources are sparsely distributed in the environment.

(v) This term summarizes the concentration distribution of the previous time step. We make use of this term since temporal interdependence between two time stamps cannot be easily resolved for the concentration distribution. As a possible solution we include this PDF that can can be computed from the posterior (4.1) obtained at the time step $n-1$ and appropriately marginalized as follows:

$$\begin{aligned}
&p(\vec{f}[n-1]|\vec{y}[0],...,\vec{y}[n-1]) = \qquad (4.2)\\
&= \int \cdots \int p(\vec{f}[n-1], \vec{u}[n-1], \vec{v}_1[n-1], \vec{v}_2[n-1], \vec{f}[n-2]|\vec{y}[0],...\\
&...,\vec{y}[n-1])d\vec{v}_1[n-1]d\vec{u}[n-1]d\vec{v}_2[n-1]d\vec{f}[n-2]
\end{aligned}$$

In this way, the estimation of the parameters of interest for a single time step can be implemented similar to a filtering procedure.

4.2 Probabilistic Gas Dispersion and Measurement Model

Let us now have a closer look at the term (ii) $p(\vec{f}[n]|\vec{u}[n], \vec{v}_1[n], \vec{v}_2[n], \vec{f}[n-1])$ of (4.1) and how to obtain this PDF. The conditional PDF represents the probability of a gas distribution $\vec{f}[n]$ given the current source distribution, wind field and the gas distribution of the previous time stamp. The deterministic equivalent to the conditional PDF is the system of algebraic equations obtained from the numerical approximation of the PDE in the previous chapter. It provides a solution for the gas distribution $\vec{f}[n]$ given the other environmental parameters by setting the residual vector $\vec{r}[n]$ to zero. In contrast to the deterministic case, where each equation has to hold exactly, i.e. $\vec{r}[n] = 0$, in a probabilistic framework we propose to relax the equations and assume that the equality holds within a certain precision. This is an important difference to classical deterministic approaches and one of the main contributions of our work. By relaxing the residual vector we are able to quantify our trust into the dispersion model. We propose that elements of the residual vector $r_c[n]$ are spatially and temporally white, normally distributed with zero mean and a precision τ_s.

We further assume that the elements $r_c[n]$ are statistically independent. This can be justified by the following consideration: Let us consider the system of equations obtained by FDM. These equations are a numerical approximation of the PDE (3.1). The PDE is valid in every infinite small point in the considered domain. Similarly, the single equation (3.11) is valid for each individual cell c. The equation holds everywhere in the environment independent where the cell is or what is going on in surrounding cells. Thus, also the residuum $r_c[n]$ of an individual cells must be independent of other residua. Therefore, in a probabilistic framework the elements of the residual vector $r_c[n]$ can be assumed as statistically independent.

To summarize, we allow the system of equations to deviate from 0, yet with an expected deviation to be zero. This permits us to define a conditional PDF (ii) in (4.1) as follows:

$$p(\vec{f}[n]|\vec{u}[n], \vec{v}_1[n], \vec{v}_2[n], \vec{f}[n-1]) \propto \prod_c e^{-\frac{\tau_s}{2} r_c[n]^2}, \tag{4.3}$$

where $\vec{r}_c[n]$ actually is a function of $\vec{f}[n], \vec{u}[n], \vec{v}_1[n], \vec{v}_2[n]$ and $\vec{f}[n-1]$ according to (3.38). The Gaussian shape arises from the fact that we have chosen $r_c[n]$ to be normally distributed with zero mean and a precision τ_s. It can be considered as an analogon to least square optimization in a deterministic framework. This thesis focuses on this special case of normal distributed residuum. Other option maybe considered in the future work.

By means of the precision τ_s the trust into the dispersion model is parametrized. In other words the parameter τ_s quantifies the uncertainty of our model assumptions. By setting $\tau_s \to \infty$, i.e. high trust or low uncertainty, the original algebraic system of equations is recovered exactly. For small τ_s we allow the dispersion process to deviate from the model assumptions and thus "tolerate" a certain model mismatch caused by dynamic effects that are not captured with an advection-diffusion PDE. This is a nice feature of our approach, since we already know that our simplified dispersion model is only an approximation of the complex dynamic gas dispersion process in reality.

So far we have not considered the boundary conditions in the conditional PDF (4.3). In order to extend (4.3), we have to translate the equations arsing from the boundary conditions (3.42) to the probabilistic framework, too. This is done in the same way as for the system of equations (3.38) and their residual vector $\vec{r}[n]$. Again we relax the residual vector $\vec{b}$ of the boundary conditions by a precision τ_b. Then the extended conditional PDF is given as

$$p(\vec{f}[n]|\vec{u}[n], \vec{v}_1[n], \vec{v}_2[n], \vec{f}[n-1]) \propto \prod_c e^{-\frac{\tau_s}{2} r_c[n]^2} \prod_c e^{-\frac{\tau_b}{2} b_c^2}, \tag{4.4}$$

where we use the precision τ_b to model the trust into the boundary conditions. Practically, a large value of τ_b would demand that the boundary constraints are fulfilled strictly, where as a low value would relax the boundary constraints.

Besides the dispersion model, we introduced a measurement model in Section 3.4 characterizing the gas sensors carried by the multi-agent system. Based on equation (3.43) and the Gaussian assumption of the measurement noise we can formulate the conditional PDF of a measurement.

$$p(\vec{y}[n]|\vec{f}[n]) \propto \prod_{l \in \mathcal{L}[n]} e^{-\frac{\tau_m}{2} \left\| y_l[n] - \vec{m}_l^T[n] \vec{f}[n] \right\|^2}. \tag{4.5}$$

Here $\mathcal{L}[n]$ is the set of robots that provide a measurement at time stamp n. The precision τ_m is the inverse variance of the measurement noise introduced in Section 3.4.

4.3 Sparsity Inducing Source Prior

An important contribution of the approach presented in this thesis is the design and choice of the prior PDF for the source distribution. Essentially, the prior has two functionalities. First it acts as a kind of regularization to our problem. As discussed in Section 3.5, the solution of the source estimation problem is ambiguous in general. However, through the design of the prior we are able to restrict the solution space and favor source distributions with certain properties. This also leads to the second role of the prior: It enables us to model possibly available a priori knowledge about the sparsely distributed source. In our case we do not know the exact number of sources but we know that the sources are sparsely distributed in the environment. By a proper design of the prior we can encode this knowledge. Here let us recap what a sparse source distribution means from a mathematical point of view. For the deterministic case (see Section 3.5) the vector $\vec{u}$ represents the spatial distribution of sources. A sparse distribution means that most of the entries in $\vec{u}$ are zero. Similarly, in the probabilistic framework a high likelihood is assigned to source distributions that are sparse. In other words the posterior PDF possesses its maximum at a constellation where most of the elements of $\vec{u}$ are zero.

In the following we illustrate the effect of the sparsity inducing prior by means of an example and we turn to the question: How can we design a prior PDF in order to induce sparsity of the source distribution?

4.3.1 Illustrating Example

To better understand the concept of sparsity introducing priors let us have a look at a simplified example for a low-dimensional space. Therefore, we assume an underdetermined linear (system) of equation(s) for two parameters u_1 and u_2 arising from a single measurement $y = 1$. The residuum r may be, for example, given by the single equation

$$r = u_1 + 2u_2 - y. \tag{4.6}$$

Analogously to the relaxation of the residuum of the gas dispersion in the previous section, we relax the residuum r due to measurement noise and model uncertainties. Thus, for our example we get a likelihood PDF:

$$p(y|u_1, u_2) \propto e^{-\frac{\tau_s}{2}(u_1+2u_2-y)^2}. \tag{4.7}$$

For the confidence into the model $\tau_s = 20$, the PDF is plotted in the first column of Figure 4.2. Now let's have a look at a few different prior distributions and their resulting posterior PDF. Gaussian Prior:

$$p(u_1, u_2) \propto \prod_i e^{-\frac{\tau_g}{2}(u_i)^2}, \quad \text{(here: } \tau_g = 20\text{)} \tag{4.8}$$

Laplace Prior:

$$p(u_1, u_2) \propto \prod_i e^{-\frac{\tau_l}{2}|u_i|} \quad \text{(here: } \tau_l = 10\text{)} \tag{4.9}$$

Student's t Prior:

$$p(u_1, u_2) \propto \prod_i \left(1 + \frac{u_i^2}{n}\right)^{-(n+1)/2} \qquad \text{(here: } n = 0.001\text{)} \tag{4.10}$$

These prior distributions are illustrated in the middle column of Figure 4.2. The resulting posteriors, i.e. $p(u_1, u_2|y) = p(y|u_1, u_2) \times p(u_1, u_2)$, are shown in the last column. Especially, the maximum of the posterior is of interest. Here we would like to remark again that a sparse solution lies either on the u_1 or u_2 axis, since then either u_2 or u_1 is zero. For our example the two desired sparse solutions that fulfill the linear equation (4.6) are: (i) $u_1 = 0, u_2 = 0.5$ or (ii) $u_1 = 1, u_2 = 0$. For the Gaussian prior finding the maximum of the posterior corresponds to a least square problem with l_2 regularization. As can be seen from the posterior in Figure 4.2, this maximum of the posterior is far away from one of the desired sparse solutions. For the Laplace prior finding the maximum of the posterior is similar to a least square problem with an l_1 regularization and can be found by a LASSO solver (see also section 3.5). In this case the maximum is close to one of the desired sparse solutions ($u_1 =$

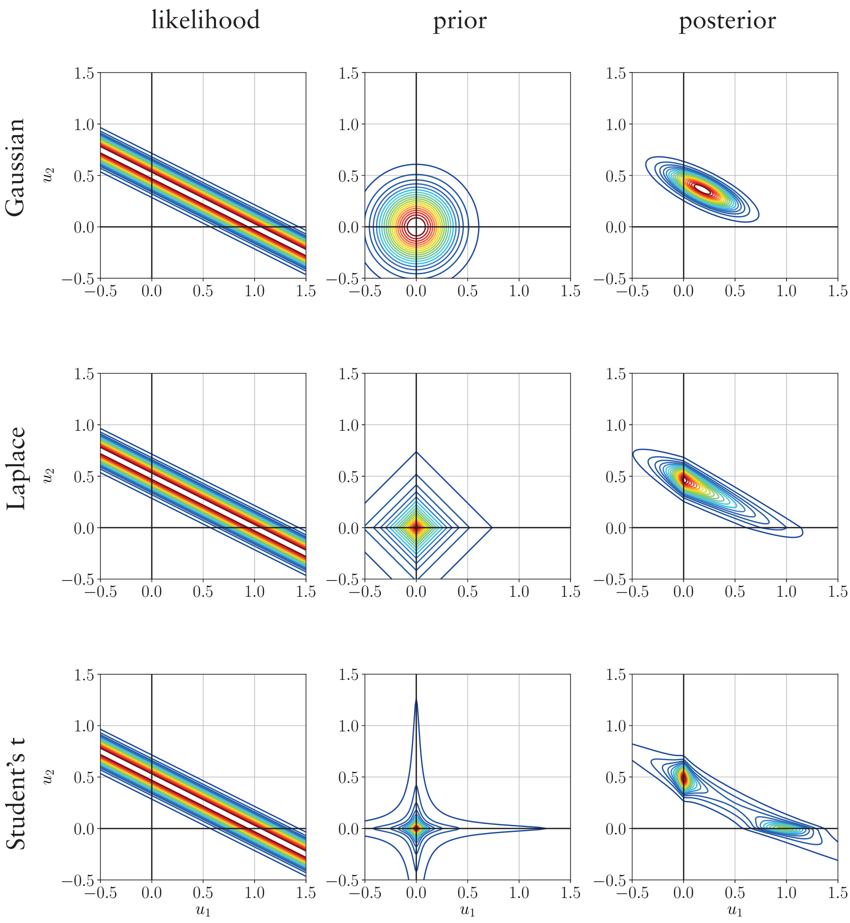


Figure 4.2: This figure illustrates the effect of different prior PDFs on the resulting posterior PDFs. The left column shows the same likelihood for all cased. In the first row a Gaussian prior is shown in the middle as well as the resulting posterior on the right. The second row shows the case of a Laplace prior and the third row the case of a Student's t prior. Only in case of the Laplace and Student's t prior, the maximum of the posterior is located on the axes. States on the axes are sparse, since either u_1 or u_2 is zero.

$0, u_2 = 0.5$). Similarly, for the Student's t prior the maximum of the posterior is close to this sparse solution. In addition, for the Student's t prior the other sparse solution at $u_1 = 1, u_2 = 0$ has a high probability. To sum up, while a Gaussian prior is not appropriate to achieve a sparse solution, the Laplace and Student's t distribution are suitable priors to incorporate sparsity assumptions in the probabilistic framework. Loosely speaking, their distributions put a lot of probability mass on the axes and thereby favor solutions that are sparse. Following this interpretation, the heavy tailed Student's t distribution seems even superior to the Laplace distribution.

4.3.2 Sparse Bayesian Learning

In order to realize a Student's t distribution we use SBL methods. SBL is realized by imposing a hierarchical factorized prior [174, 195]

$$p(\vec{u}[n], \vec{\gamma}[n]) = \prod_{c=1}^{C} p(u_c[n]|\gamma_c[n])p(\gamma_c[n]). \tag{4.11}$$

Here we introduced a new hyper-parameter $\gamma_c[n]$ for each element in $\vec{u}[n]$. The hyper-parameters $\gamma_c[n]$ are also treated as a random variable and has to be estimated as well as $\vec{u}[n]$. The product $p(u_c[n]|\gamma_c[n])p(\gamma_c[n])$ builds a so called Gaussian scale mixture, where $p(u_c[n]|\gamma_c[n])$ is a zero-mean Gaussian PDF with precision $\gamma_c[n]$ ($p(u_c[n]|\gamma_c[n]) = N(u_c[n]|0, \gamma_c[n]^{-1})$). The hyper-prior $p(\gamma_c[n])$ is selected as a Gamma distribution $G(\gamma_c[n]|a, b)$, $c = 1, \ldots, C$ with fixed hyper-prior parameters a and b. It was shown in [174] that when marginalizing over γ_c, we obtain

$$\begin{aligned} p(u_c[n]) &= \int p(u_c[n]|\gamma_c[n])p(\gamma_c[n])d\gamma_c[n] = \\ &= \int N(u_c[n]|0, \gamma_c[n]^{-1})G(\gamma_c[n]|a, b)d\gamma_c[n] = \\ &= \frac{\Gamma(\frac{\nu+1}{2})}{\sqrt{\nu\pi}\Gamma(\frac{\nu}{2})}\left(1 + \frac{u_c[n]^2}{\nu}\right)^{-\frac{\nu+1}{2}}, \text{with} \quad \Gamma(x) = \int_0^{\text{inf}} t^{x-1}e^{-t}dt. \end{aligned} \tag{4.12}$$

Since the Gamma distribution is a conjugate prior of the Gaussian distribution with fixed mean, i.e. zero, and unknown precision, i.e. γ_c, the marginalization has a closed form solution, which turns out to be the Student's t distribution with the degree of freedom ν depending on a and b.

Finally, here we will make use of a popular version of SBL that uses the non-informative hyper-prior $p(\gamma_c[n]) \propto \gamma_c[n]^{-1}$ obtained when $a_\gamma \to 0$ and $b_\gamma \to 0$ [194, 195, 174]. The motivation for this choice is twofold. First, the resulting inference schemes typically demonstrate superior (or similar) performance as compared to schemes derived based on other hyper-prior selections

[67]. Second, very efficient inference algorithms can be constructed and studied [175, 158, 194, 72].

4.4 Air Flow Prior

In the probabilistic approach introduced in Section 4.1 we need a prior PDF $p(\vec{v}_1[n], \vec{v}_2[n])$ for the two components $\vec{v}_1[n], \vec{v}_2[n]$ of the airflow vector for each point in space and every time stamp n. In the following we call this PDF a probabilistic airflow model. Such kinds of models are common in meteorology or atmospheric research [57]. However, in this field the models aim to describe the airflow on a macro ($> 100km$) or meso ($1km - 100km$) scale. For robotic source localization we rather require a micro scale airflow model for a region smaller than $1km^2$. For this scale CFD models are commonly used [122]. As discussed in Section 3.1, the main disadvantage of these models is their high computational costs. Therefore, we follow the data driven modeling approach inspired by [27], but with two simplifications:

(i) we do not distinguish between laminar and turbulent components in the extrapolation (essentially the main contribution of [27]) and

(ii) instead of using a histogram based representation of PDFs as in [27], we use approximating closed form representations of the probabilistic distributions.

Commonly, analytic probabilistic descriptions of airflow or wind make use of a Weibull distribution [40] for the absolute wind velocity and a von-Mises distribution for the wind direction [39, 26]. For computational simplicity we prefer a PDF (or joint PDF) of the two wind velocity components in a Cartesian coordinate system and propose to use two Normal distributions for the wind velocity components that are statistically independent. This choice is mostly motivated by simpler and analytically tractable calculations. However, as we will show in Section 7.3.2, measurements also support this assumption. Consequently, the airflow model is essentially a spatial map providing the mean and variance for the two velocity components (in a Cartesian coordinate system) at each point in space and for every time stamp. This map is build based on wind measurements. Since, during the exploration procedure we do not have wind measurements for all points in space at the same time, we need a way to interpolate in between available measurements and thereby predict the wind velocity at all locations.

Let us assume we have taken wind measurements $y_{v1,i}, y_{v2,i}$ for the two wind components at the time stamp n_i at a location $\vec{x}_i \in \Omega$. In general multiple measurements ($i = 0, 1, 2, ...$) are available. For the current time stamp n we can calculate the probabilistic airflow map as follows: For all points $\vec{x}_k$ of our airflow map the predicted mean values are

$$\mu_{v1,k}[n] = \frac{\sum_i w(\vec{x}_k, n, \vec{x}_i, n_i) y_{v1,i}}{\sum_i w(\vec{x}_k, n, \vec{x}_i, n_i)} \tag{4.13}$$

$$\mu_{v2,k}[n] = \frac{\sum_i w(\vec{x}_k, n, \vec{x}_i, n_i) y_{v2,i}}{\sum_i w(\vec{x}_k, n, \vec{x}_i, n_i)}, \tag{4.14}$$

where $w()$ is a Gaussian shaped weighting function of the observations

$$w(\vec{x}_k, n, \vec{x}_i, n_i) = e^{-\frac{2}{k_s}\|\vec{x}_i - \vec{x}_k\|^2} e^{-\frac{2}{k_t}(n_i - n)^2}. \tag{4.15}$$

Loosely speaking, an observation is down weighted when the observation location $\vec{x}_i$ is further away from $\vec{x}_k$ and when the observation time stamp n_i was further in the past. The shape of the Gaussian weighting function is parametrized by the kernel widths k_s and k_t. The choice of these parameters is further investigated in Section 7.3.2. Further, we can calculate the variances at all points $\vec{x}_k$ of our airflow map as:

$$\sigma^2_{v1,k}[n] = \frac{\sum_i w(\vec{x}_k, n, \vec{x}_i, n_i)(y_{v1,i} - \mu_{v1,k}[n])^2}{\sum_i w(\vec{x}_k, n, \vec{x}_i, n_i)} \tag{4.16}$$

$$\sigma^2_{v2,k}[n] = \frac{\sum_i w(\vec{x}_k, n, \vec{x}_i, n_i)(y_{v2,i} - \mu_{v1,k}[n])^2}{\sum_i w(\vec{x}_k, n, \vec{x}_i, n_i)}. \tag{4.17}$$

Please note that this interpolation technique is quite similar to the Kernel DM+V algorithm [104]. However, here we apply the technique to the wind velocity components instead of the gas concentration as was done in [104]. The down weighting of measurements further in the past is adopted from the work in [22] By evaluating the means and variances at the cells or nodes c of our discretized environment we get the probabilistic airflow map:

$$p(\vec{v}_1[n], \vec{v}_2[n]) = \tag{4.18}$$

$$\prod_c^C N([\vec{v}_1]_c[n] | \mu_{v1,c}[n], \sigma^2_{v1,c}[n]) \cdot \prod_c^C N([\vec{v}_2]_c[n] | \mu_{v2,c}[n], \sigma^2_{v2,c}[n]).$$

(Note: here $[\vec{v}_1]_c$ indicates the cth element of the vector $\vec{v}_1$). This PDF acts as our prior distribution of the air flow or wind and can be directly plugged into the probabilistic formulation (4.1).

4.5 Summary

In this chapter we made use of the deterministic gas dispersion model developed in the previous chapter and translated it into a Bayesian probabilistic framework. More precisely, we formulated the posterior PDF for the gas concentration, source distribution and wind field. The posterior PDF factorizes into different parts: the likelihood makes use of the measurement model, the gas dispersion model is used to problematically model transition from time stamp $n-1$ to n and prior distributions account for a-priori knowledge. For the design of source prior we apply Sparse Bayesian Learning (SBL) techniques to incorporate the prior assumption that sources are sparsely distributed in our environment. As a prior for the wind field, we developed a data driven probabilistic airflow map that can be generated before the exploration procedure or in parallel to the exploration by means of wind measurements. Finally, let us remark that so far we only formulated the posterior PDF. In a next step we have to "solve" this PDF, i.e. to calculate parameters and features of interest from the PDF.

Chapter 5
Implementation for Multi-Robot Applications

In the previous section we cast gas source localization into a probabilistic estimation problem and formulated the posterior PDF for the spatial gas distribution, the spatial source distribution and the wind field given our gas concentration measurements. Now we are interested in certain features of this posterior PDF, for example its maximum. This maximum tells us which combination of gas and source distribution has the highest probability for the given observations. Actually, this maximum can be considered as the solution to our inverse problem: the estimated source distribution given concentration measurements. Besides the maximum, we are interested in the second central moment of the posterior, i.e. the variance. This second moment will be used later on as a gauge for the information content of different regions in the exploration environment. It will be the foundation of the DARES strategy designed in section 6.2. Further, we need to calculate marginal distributions of the posterior, since we are interested in the PDF of the source distribution in a individual single grid cell but not in the whole domain. In this chapter we therefore investigate different algorithms to calculate such features of the posterior. The chapter is divided into two main sections. In the first section we have a look at the simpler case of a linear dispersion model and how we can profit from a multi-robot system and a distributed implementation. In the second section we will consider the more complex case of a non-linear dispersion model.

5.1 Distributed Implementation of the Linear Model

In this section we study the case of a linear dispersion model for a process purely driven by diffusion, i.e. absence of air flow). This means the residual vector $\vec{r}[n]$ in equation (4.4) and accordingly in the posterior (4.1) is calculated based on the linear equations (3.40). Further, for ease of interpretation, we assume the equations are constructed using FDM and the environment is discretized into grid cells as explained in section 3.3.1. In case of FEM the presented approach will work in a similar way. However, the mathematical formulation will be more involved. This section develops a distributed inference algorithm to calculate the marginal PDF of the posterior as well as their mean and variances. The algorithm also accounts for the sparsity inducing prior of the source distribution.

5.1.1 Factor Graphs Representation

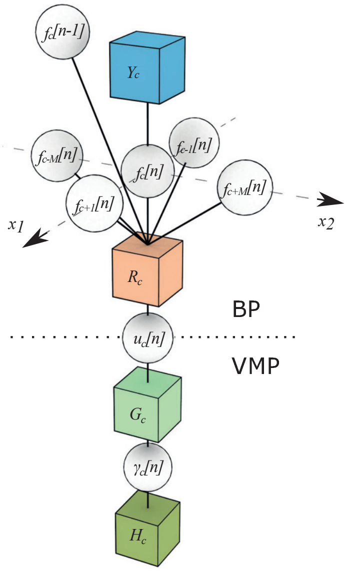


Figure 5.1: This factor graph represents a single grid cell in the posterior PDF. Cubes denote factor nodes, while spheres stands for variable nodes. The graph could be defined into two parts (BP and VMP), which will get important in Section 5.1.2.

The inference approach that we consider here is based on representing the posterior (4.1) using a factor graph [88]. A factor graph is an undirected bipartite Bayesian network being composed of value nodes, which represent random variables, and factor nodes, which model functional dependencies between them. The origin of the factor graph is in coding theory [170, 61], where it is used for error detection and correction. However, it is a powerful framework for general probabilistic modeling. The factor graph is used as a graphical representation of relationships between different random variables. Concerning a factorized form of a PDF, each factor node in the graph corresponds to a factor of the product in the posterior (4.1) (See [108] for a more detailed introduction to factor graphs). In our case we profit from the already nicely factorized form of the posterior (4.1). Note that not only (4.1) is a product, but also the factors of the product themselves are factorized e.g. (4.4) or (4.11). Let us shortly recap the FDM of section 3.3.1. Essentially, each line of the system of equations we gain from the FDM according to (3.10) puts the concentration value $\vec{f}_c[n]$ and the source strength $\vec{u}_c[n]$ in a grid cell c into a mathematical relation with the concentration values $\vec{f}_{c+1}[n], \vec{f}_{c-1}[n], \vec{f}_{c+J}[n], \vec{f}_{c-J}[n]$ of the neighboring grid cells and the concentration value of the previous time stamp $\vec{f}_c[n-1]$. With this insight let's have a look at the factor graph of a single grid cell shown in Figure 5.1 The graph shows factor nodes as cubes and vari-

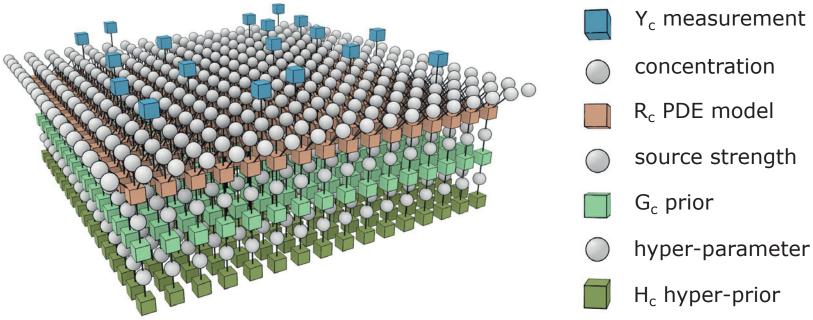


Figure 5.2: This simplified factor graph illustrates how the graphical representation of the whole posterior PDF looks like. For simplification time dependencies and boundary conditions are neglected. The blue cubes represent measurements. They only exist at cells where in this example a measurement was actually taken.

able nodes as spheres for a single grid cell and its direct neighbors are shown. The factor nodes in the graph put all connected value nodes into a functional relationship. The relationship of the neighboring concentration values and the source strength given by our PDE model is represented by the factor node R_c. Further, a factor node Y_c represents the state likelihood of a single grid cell arising from a measurement as given by (4.5). Note that this node exists only when the measurement at the associated grid cell is available. Moreover, the factors G_c and H_c capture the hierarchical source prior (4.12) of the sparsity assumption. Here, G_c represents the parametrized prior $p(u_c[n]|\gamma_c[n])$, where H_c represents the hyper prior PDF $p(\gamma_c[n])$

Based on the graph for a single grid cell as shown in Figure 5.1, we can design the graph for the combination of all grid cells in the domain. This overall graph is shown in Figure 5.2. To simplify the graph in order to make a visual representation possible, time dependencies to previous time stamps are not shown. In practice, the whole graph as shown in Figure 5.2 would exist for the previous time stamp, too, and all concentration nodes of both graphs are connected. Furthermore, the constraints or relations given by the boundary conditions are not shown for simplicity. The boundary conditions would add additional factor nodes to the graph which are connected to the concentration nodes on the border.

At this point let us remark on the structure of the factor graph in the context of the chosen numerical approximation of the PDE. The graph in Figure 5.1 is specific for the chosen form of the discretization. Especially, the part connected to the factor node R_c is closely related to the used FDM. In case of other approximation schemes or other PDEs, the graph will have a slightly different topology. Yet the graph design for another numerical approximation is still straight forward and follows the same steps. Loosely speaking, the connectivity of the graph and its adjacency matrix is given by the system matrix of linear equations arising from the approximation scheme. Moreover, for a diffusion PDE the parts in the graph related to the prior and the measurement nodes are the same independent of the numerical approximation of the PDE. In other words, the numerical approximation of the PDE only affects the nodes R_c at their connection, while the rest of the graph remains the same.

5.1.2 Inference via Message Passing

The purpose of the factor graph in Figure 5.2 is to facilitate inference of the concentration distribution $\vec{f}[n]$, source distribution $\vec{u}[n]$, and the corresponding hyper-parameters $\vec{\gamma}[n]$. In this thesis inference is performed using message passing on factor graphs [88], [108]. Message passing algorithms are a powerful tool to calculate marginal PDFs of random variables in general. In a nutshell, the message passing algorithms work as follows. Messages are exchanged between nodes of the factor graph along its edges, i.e. from factor nodes to variable nodes and from variable nodes to factor nodes. The messages are PDFs themselves (sometimes also referred to as beliefs). Outgoing messages from a node are calculated based on the incoming messages at a node. Message passing algorithms provide update rules how to calculate these outgoing messages. By interactively exchanging the messages between nodes the outgoing messages of the variable nodes converge to the marginal PDF of the corresponding random variable. A nice property of these algorithms is that they can be implemented in a distributed fashion very easily. Thus, they are particularly useful for a multi-agent system. Note that the following derivation of the message passing algorithm was first published in [189].

Before we begin with the detailed description of the message update rules, let us have a look at the shape of the individual factors of the posterior (4.1). For the linear case considered here (without wind) all messages are Gaussian except from the messages connected to the sparsity inducing prior. This has an important practical consequence. We can divide the graph into two parts as shown in Figure 5.1. For the part above the dotted line, the estimation can be efficiently solved via a standard sum-product algorithm (or Belief Propagation (BP)) [135]. Moreover, all corresponding messages can be expressed in closed form. For the lower part, however, which is responsible for sparsity constraints, applying the sum-product algorithm [135] will lead to analytically intractable messages. To circumvent this we will approximate the resulting messages in the lower part using variational Bayesian techniques. Specifically, we follow the approach proposed in [146], where the authors applied Variational Message Passing to some parts of the factor graph in order to compute analytical approximations of the messages, whereas the rest of the graph is solved by the sum-product algorithm.

The sum-product algorithm

The sum-product algorithm [135],[88],[108], also known as belief propagation (BP), computes marginal distributions of value nodes by exchanging messages – functions (or rather PDFs) of graph variable nodes – between the nodes along the graph edges. In cases, where a factor graph is free of loops, all messages have to be sent only once in order to calculate the exact marginal distributions. For graphs that contain loops, as in our case due to the spatio-temporal neighbourhood relations, the sum-product algorithm is not guaranteed to converge. Yet in many practical cases it is known that loopy belief propagation [60], when messages are transmitted several times in the graph, does converge to an approximative solution [127]. As we will show later in Section 5.1.3, this is also the case for our graph topology and parametrization.

In the following, we will derive all messages that are needed for inference on the factor graph. To this end, we make use of the following notation. The notation $vec(A)$ arranges all elements of the set A in a vector. Similar the notation $diag(A)$ generates a diagonal matrix with the elements of A. The statement $m_{f_c[n]\rightarrow R_c}$ represents the PDF – a message – sent from a variable node $f_c[n]$ to factor node R_c. As will become apparent, most of the messages will be Gaussian functions $N(x|\mu, \tau^{-1})$, which simplifies their exchange between graph nodes. Gaussian messages are parametrized by the message mean μ and precision τ, and thus only these parameter values need to be communicated. We will use $\mu_{f_c[n]\rightarrow R_c}$ to represent the mean of a Gaussian message sent from node $f_c[n]$ to node R_c, and $\tau_{f_c[n]\rightarrow R_c}$ to represent the corresponding message precision. In contrast, non-Gaussian messages generally require an appropriate numerical representation of the PDF (e.g., a discretization) to ensure that messages can be exchanged between the nodes. Also, note that messages going out from a node depend naturally on the incoming messages, for example following the BP algorithm [135]. Thus, due to the presence of loops, the derivation below contains references to messages that are defined later in the text.

We begin with the message $m_{Y_c\rightarrow f_c[n]}$ sent from the top factor Y_c to the corresponding variable node $f_c[n]$ (c.f. Fig. 5.1). Due to the structure of the graph, $m_{Y_c\rightarrow f_c[n]}$ is merely the likelihood of the state in cell c given by (4.5):

$$m_{Y_c\rightarrow f_c[n]} = N(f|\mu_{Y_c\rightarrow f_c[n]}, \tau_m^{-1}), \tag{5.1}$$

with $\mu_{Y_c\rightarrow f_c[n]}$ being the actually measured value $y_c[n]$ in cell c at time n.[1] Here, the precision τ_m corresponds to the measurement noise. Note that this

[1] Note that this is true provided that $\vec{M}[n]$ is a selection matrix, i.e., when $y_c[n]$ is a noisy version of $f_c[n]$.

message is sent (computed) only when the corresponding measurement at the cell c at time n has been made. Otherwise the node Y_c would not exist (see Figure 5.2).

The computation of messages sent from the variable nodes $f_c[n]$, $f_{c-1}[n]$, $f_{c+1}[n]$, $f_{c-J}[n]$, $f_{c+J}[n]$, $f_c[n-1]$ toward R_c is a bit more involved but still straightforward. Due to the assumed probabilistic model, the message is in fact a product of several normal distributions. With the auxiliary neighborhood set $\mathcal{N}_c = \{c, c-1\ , c, c-J,\ c+J\}$, we can write

$$m_{f_j \to R_c} = m_{Y_j \to f_j[n]} \prod_{i \in \mathcal{N}_c \setminus j} m_{R_i \to f_j[n]} \propto N(f|\mu_{f_j[n] \to R_c}, \tau^{-1}_{f_j[n] \to R_c}), \quad j \in \mathcal{N}_c \tag{5.2}$$

The parameters of the normal distributions can be calculated as

$$\begin{aligned} \tau_{f_j[n] \to R_c} =& \tau_m + \sum_{i \in \mathcal{N}_c \setminus j} \tau_{R_i \to f_j[n]}, \quad \text{and} \\ \mu_{f_j[n] \to R_c} =& \frac{1}{\tau_{f_j[n] \to R_c}} \Big[\sum_{i \in \mathcal{N}_c \setminus j} \mu_{R_i \to f_j[n]} \tau_{R_i \to f_j[n]} + \mu_{Y_j \to f_j[n]} \tau_m \Big], \\ & j \in N(c). \end{aligned} \tag{5.3}$$

The update rule for messages sent from node R_c can also be computed in closed form. There are seven different types of messages depending on their destination node. We define the set $X = \{f_c[n],\ \ f_{c-1}[n], f_{c+1}[n], f_{c-J}[n], f_{c+J}[n], f_c[n-1],\ u_c[n]\}$ containing the seven different destination nodes. In order to calculate all outgoing messages $m_{R_c \to j}, j \in X$ we define auxiliary variables, vectors and matrices to simplify the derivation. Let's recap the definition of the residual vector $\vec{r}[n]$ for a grid cell for the linear diffusion equation (see equation (3.10) and (3.40)). We can write

$$r_c[n] = \sum_{i \in X} a_i i \quad , \text{ with} \tag{5.4}$$

$$a_{f_c[n]} = \frac{4}{\rho^2} + \frac{1}{\Delta t}, \quad a_{f_{c-1}[n]} = \frac{-1}{\rho^2}, \quad a_{f_{c+1}[n]} = \frac{-1}{\rho^2}, \quad a_{f_{c-J}[n]} = \frac{-1}{\rho^2},$$
$$a_{f_{c+J}[n]} = \frac{-1}{\rho^2}, \quad a_{f_c[n-1]} = -\frac{1}{\Delta t}, \quad a_{u_c[n]} = -1.$$

Further, for the calculation of the message towards $j \in X$ we introduce

$$\begin{aligned} \vec{s}_j =& vec(\{i : i \in X \setminus j\}) \\ \vec{\mu}_j =& vec(\{\mu_{i \to R_c} : i \in X \setminus j\}) \\ \mathbf{T}_j =& diag(\{\tau i \to R_c : i \in X \setminus j\}) \\ \vec{\tilde{a}}_j =& vec(\{a_i : i \in X \setminus j\}) \end{aligned} \tag{5.5}$$

Using these definitions and standard rules for computing messages from a factor node to a value node [88], we obtain for the message from R_c towards a variable node $j \in X$:

$$
\begin{aligned}
m_{R_c \to j} &\propto \int e^{-\frac{\tau_s}{2}(r_c[n])^2} \times \prod_{i \in X \setminus j} m_{i \to R_c} d\vec{s}_j \\
&\propto \int e^{-\frac{\tau_s}{2}(\vec{\hat{a}}_j^T \vec{s}_j - a_j j)^2} \times e^{-(\vec{s}_j - \vec{\mu}_j)^T \vec{T}_j (\vec{s}_j - \vec{\mu}_j)} d\vec{s}_j,
\end{aligned}
\tag{5.6}
$$

Since the integrand in (5.6) is a product of Gaussians, i.e. again a Gaussian , the message $m_{R_c \to j}$ can be computed exactly. After some algebra, we get

$$
m_{R_c \to j} = N(u | \mu_{R_c \to j}, \tau_{R_c \to j}^{-1}),
\tag{5.7}
$$

with the mean and the precision given as

$$
\begin{aligned}
\tau_{R_c \to j} =& \tau_s a_j^2 - \tau_s^2 \vec{\hat{a}}^T a_j^2 (\vec{T}_j + \tau_s \vec{\hat{a}}_j \vec{\hat{a}}_j^T)^{-1} \vec{\hat{a}}_j \\
=& a_j^2 \left(\tau_s^{-1} + \vec{\hat{a}}_j^T \vec{T}_j^{-1} \vec{\hat{a}}_j \right)^{-1} \\
\mu_{R_c \to j} =& \frac{\tau_s a_j \vec{\mu}_j^T \vec{T}_j (\vec{T}_j + \tau_s \vec{\hat{a}}_j \vec{\hat{a}}_j^T)^{-1} \vec{\hat{a}}_j}{\tau_{R_c \to j}} \\
=& \frac{1}{a_j} \vec{\mu}_j^T \vec{\hat{a}}_j,
\end{aligned}
\tag{5.8}
$$

with the final expressions having been obtained by applying the Sherman-Morrison formula [23, 68] to $(\vec{T}_j + \tau_s \vec{\hat{a}}_j \vec{\hat{a}}_j^T)^{-1}$.

Unfortunately, the other messages in the factor graph cannot be computed in closed from. The reason for that is the hyper-parameter γ with its Gamma shaped PDF. In order to efficiently represent all intractable messages, we approximate them using a class of analytical approximation tools. Specifically, we use variational Bayesian techniques as discussed in the following.

Variational Message Passing (VMP)

While in general the loopy BP algorithm is applicable to arbitrary factor graphs, the analytical tractability of the messages is often limited. For example messages sent from factor nodes require a marginalization (integration), which may be intractable for non-linear factor nodes. In our case, messages related to the sparsity prior lead to non-Gaussian PDFs, which also prohibits exact evaluation of the messages. Bayesian variational methods, and in particular VMP [193], are a class of analytical approximation techniques that allow circumventing the intractability of message computations.

In VMP, the belief distributions are approximated with simpler messages, typically chosen from a certain parametric family to minimize a Kullback-Leibler divergence [90] between the true beliefs and an approximation. Naturally, message computation rules for VMP differ from those computed with BP (see [193] for more details). In what follows, we give a summary of the corresponding computations specifying the required messages. In our case, we need to compute messages that propagate between variable node $u_c[n]$ and factor G_c (see also Figure 5.1):

$$m_{u_c[n]\to G_c} \propto m_{G_c\to u_c[n]} m_{R_c\to u_c[n]}, \quad \text{and} \tag{5.9}$$

$$m_{G_c\to u_c[n]} \propto \exp \int \ln N(u_c[n]|0, \gamma_c[n]^{-1}) \times m_{\gamma_c[n]\to G_c} d\gamma_c[n]; \tag{5.10}$$

between variable node $\gamma_c[n]$ and factor G_c:

$$m_{\gamma_c[n]\to G_c} \propto m_{G_c\to\gamma_c[n]} m_{H_c\to\gamma_c[n]}, \quad \text{and} \tag{5.11}$$

$$m_{G_c\to\gamma_c[n]} \propto \exp \int \ln N(u_c[n]|0, \gamma_c[n]^{-1}) \times m_{u_c[n]\to G_c} du_c[n]; \tag{5.12}$$

and the message between $\gamma_c[n]$ and factor H_c

$$m_{H_c\to\gamma_c[n]} \propto 1/\gamma_c[n]. \tag{5.13}$$

Finally, the connection between VMP and BP is the message

$$m_{u_c[n]\to R_c} \propto m_{G_c\to u_c[n]} m_{R_c\to u_c[n]}. \tag{5.14}$$

Fixed point of Variational Message Passing (VMP)

Since our graph contains loops, we have to send these messages several times until convergence. Instead of performing multiple iterations of message updates in the whole graph, we propose first to perform updates of the VMP part of the graph by fixing messages $m_{R_c\to u_c[n]}$. In other words we keep all part, i.e. messages, belonging to the BP in Figure 5.1 fixed and only update VMP messages. After convergence of the VMP messages, the resulting messages are propagated into the BP part. This approach is largely inspired by [159]. In fact the fixed points of the VMP messages performed ad infinitum can be computed in closed form. The corresponding analysis not only accelerates the convergence of VMP, but also reveals a condition under which the corresponding sparsity parameter $\gamma_c[n]$ is finite. It is this condition that corresponds to the detection of a source signal. Let us now study this in more detail.

Consider $m_{G_c\to u_c[n]}$ at a first update cycle $k = 0$:

$$m^{[k=0]}_{G_c\to u_c[n]} = N(u|0, (\gamma_c^{[k=0]})^{-1}), \tag{5.15}$$

which is just the prior for the source signal at the first iteration. We initialize the hyper parameter $\gamma^{[k=0]}$ by a small value like 10^{-3}. Here, the superscript k is the update iteration counter. Now, consider the computation of $m_{u_c[n]\rightarrow G_c}$ from (5.9). Note that the messages $m_{R_c\rightarrow u_c[n]}$ are proportional to a Gaussian PDF

$$m_{R_c\rightarrow u_c[n]} \propto N(u|\mu_c, \tau_c^{-1}), \tag{5.16}$$

where we define

$$\begin{aligned} \tau_c =& \tau_{R_c\rightarrow u_c[n]}, \quad \text{and} & (5.17)\\ \mu_c =& \mu_{R_c\rightarrow u_c[n]} & (5.18)\end{aligned}$$

to keep the notation short. By plugging (5.15) and (5.16) in (5.9) we get

$$m^{[k=0]}_{u_c[n]\rightarrow G_c} \propto N\left(u\middle|\frac{\mu_c\tau_c}{\tau_c+\gamma_c^{[k=0]}}, (\tau_c+\gamma_c^{[k=0]})^{-1}\right). \tag{5.19}$$

The message $m_{G_c\rightarrow\gamma_c[n]}$ to the node $\gamma_c[n]$ can be computed in closed form, because (5.12) is Gaussian. The resulting distribution can be recognized as a Gamma distribution with the corresponding PDF

$$m^{[k=0]}_{G_c\rightarrow\gamma_c[n]} \propto G\left(\gamma\middle|\frac{3}{2}, \frac{1}{2}\frac{\mu_c^2\tau_c^2+\tau_c+\gamma_c^{[k=0]}}{\left(\tau_c+\gamma_c^{[k=0]}\right)^2}\right). \tag{5.20}$$

Likewise, the message $m_{\gamma_c[n]\rightarrow G_c}$ in (5.11), which now accounts for the hyperprior H_c, is also a Gamma PDF given by:

$$m^{[k=0]}_{\gamma_c[n]\rightarrow G_c} = G\left(\gamma\middle|\frac{1}{2}, \frac{1}{2}\frac{\mu_c^2\tau_c^2+\tau_c+\gamma_c^{[k=0]}}{\left(\tau_c+\gamma_c^{[k=0]}\right)^2}\right). \tag{5.21}$$

This ends one update cycle of the VMP. At iteration $k = 1$, an updated version of the message $m_{G_c\rightarrow u_c}$ in (5.10) will take the form

$$m^{[k+1]}_{G_c\rightarrow u_c[n]} = N\left(u\middle|0, \frac{\mu_c^2\tau_c^2+\tau_c+\gamma_c^{[k=0]}}{\left(\tau_c+\gamma_c^{[k=0]}\right)^2}\right). \tag{5.22}$$

When comparing (5.22) and (5.15), it becomes apparent that the variance of the message $m^{[k+1]}_{G_c\rightarrow u_c[n]}$ becomes $\frac{\mu_c^2\tau_c^2+\tau_c+\gamma_c^{[k=0]}}{\left(\tau_c+\gamma_c^{[k=0]}\right)^2}$. By induction we can show that for $k > 0$ the following holds:

$$\gamma_c^{[k+1]} = \left(\mu_c^2\tau_c^2+\tau_c+\gamma_c^{[k]}\right)\left(\tau_c+\gamma_c^{[k]}\right)^{-2}. \tag{5.23}$$

Instead of actually sending the messages in the graph, it is more reasonable to proceed with updating the messages (5.9)-(5.12), and study the behaviour of $\Delta\gamma_c^{[k+1]} = \gamma_c^{[k+1]} - \gamma_c^{[k]}$ as $k \to \infty$, which is governed by equation (5.23). The stationary point γ^* of (5.23) is found by solving

$$\gamma_c^* = \left(\mu_c^2\tau_c^2 + \tau_c + \gamma_c^*\right)\left(\tau_c + \gamma_c^*\right)^{-2}, \tag{5.24}$$

which after some straight forward manipulations, can be found at

$$\gamma_c^* = \frac{\tau_c}{\tau_c\mu_c^2 - 1}. \tag{5.25}$$

To prove the stability of the solution (5.25) we examine γ_c^* using the Lyapunov criterion. To this end we define a Lyapunov function $\nu(\gamma_c^{[k]}) = \frac{1}{2}(\gamma_c^{[k]} - \gamma_c^*)^2$ and investigate the behavior of

$$\Delta\nu(\gamma_c^{[k]}) = \nu(\gamma_c^{[k+1]}) - \nu(\gamma_c^{[k]}). \tag{5.26}$$

By plugging (5.23) and (5.25) in (5.26) we get

$$\Delta\nu(\gamma_c^{[k]}) = -\frac{(\gamma_c^{[k]}\tau\mu_c^2 - \gamma_c^{[k]} - \tau_c)^2(\tau_c^2\mu_c^2 + 2\gamma_c^{[k]} + 3\tau_c)\tau_c}{2(\mu_c^2\tau_c^2 + \gamma_c^{[k]} + \tau_c)^2(\mu_c^2\tau_c - 1)}. \tag{5.27}$$

From the Lyapunov theory we know that the point γ_c^* is stable if $\Delta\nu \leqslant 0$, which is true when $\mu_c^2\tau_c > 1$.[2] To demonstrate the behaviour of the Lyapunov function, we evaluate $\Delta\nu(\gamma_c)$ and plot it in Fig. 5.3 for $\mu_c = 1$ and different values of τ_c. Observe that when $\tau_c\mu_c^2 > 1$, $\Delta\nu(\gamma_c)$ is negative-definite, which ensures convergence to the stationary point γ_c^*. Otherwise, γ_c^* becomes unstable in the sense that $\gamma_c^{[k]}$ will diverge from the fixed point towards positive infinity. In case that $\tau_c\mu_c^2 = 1$, the update iteration is mathematically stable according to the Lyapunov criterion, however the stationary point in (5.25) becomes infinite.

This observation allows us to define the final update rule for the message $m_{u_c[n]\to R_c}$ in (5.16), which is transmitted from the source node towards the factors R_c; it is this message that effectively combines BP and VMP inference on the graph. Using (5.22), where the variance is computed using (5.25), we get

$$m_{u_c[n]\to R_c} = \begin{cases} N(u_c|\frac{\mu_c^2\tau_c - 1}{\mu_c\tau_c}, (\tau_c + \frac{\tau_c}{\tau_c\mu_c^2 - 1})^{-1}); & \tau_c\mu_c^2 - 1 > 0 \\ N(u_c|0, \infty^{-1}) = \delta(u_c); & \tau_c\mu_c^2 - 1 \leqslant 0. \end{cases} \tag{5.28}$$

Table 5.1 summarizes all messages introduced in this section. We would like to point out that due to the proposed VMP iterations the corresponding messages are in fact not calculated. Instead, the outgoing message from u_c is

[2] Note that $\tau_c > 0$

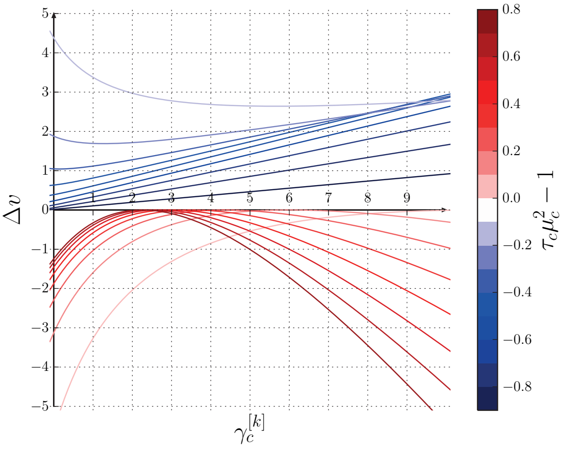


Figure 5.3: Stability analysis of the hyper-parameter γ_c: the evolution of the Lyapunov function Δv has to be negative semidefinite (here $\leqslant 0$), otherwise the update rule for the hyper-parameter is not stable (blue). This is only the case when the denominator of equation (5.25) is greater than zero.

directly computed in closed form according to (5.28). The BP messages, on the other hand, are calculated iteratively until some convergence criterion is met, or until a certain number of iterations is reached. In our case the updates are stopped after a certain number of iterations (see Section 5.1.3). We use a serial scheduling for the message passing updates. Consequently, all messages are successively calculated once per iteration of the algorithm and in a random order. Moreover, we simplify the algorithm by propagating messages only forward in time. Such an assumption is not uncommon (see [120]) and significantly reduces the complexity of the message passing algorithm. In particular, the messages $m_{R_c \to f_c[n-1]}$ are discarded. Consequently, the concentration and source distribution of the current time stamp only depends on the last time stamp. Thus, the computational complexity does not increase with respect to time and remains the same for all time stamps. This simplification is practically very reasonable, since we are often interested in an online estimation of our current states and sources for the exploration task. Improving the estimation of states or sources lying in the past is not directly helpful in our case.

Since the complexity of the proposed message passing algorithm is proportional to the number of messages, the complexity of one update iteration is $\mathcal{O}(C + M[n])$, where C is the number of cells and $M[n]$ the number of measurement taken at time stamp n. (For the static case $M[n]$ is the total

Message	Equation
$m_{Y_c \to f_c[n]}$	(5.1)
$m_{R_c \to f_c[n]}$	(5.7) with $j = f_c[n]$
$m_{R_c \to u_c[n]}$	(5.7) with $j = u_c[n]$
$m_{R_c \to f_{c-1}[n]}$	(5.7) with $j = f_{c-1}[n]$
$m_{R_c \to f_{c+1}[n]}$	(5.7) with $j = f_{c+1}[n]$
$m_{R_c \to f_{c-J}[n]}$	(5.7) with $j = f_{c-J}[n]$
$m_{R_c \to f_{c+J}[n]}$	(5.7) with $j = f_{c+J}[n]$
$m_{f_c[n] \to R_c}$	(5.3) with $j = c$
$m_{f_{c-1}[n] \to R_c}$	(5.3) with $j = c + 1$
$m_{f_{c+1}[n] \to R_c}$	(5.3) with $j = c - 1$
$m_{f_{c-J}[n] \to R_c}$	(5.3) with $j = c + J$
$m_{f_{c+J}[n] \to R_c}$	(5.3) with $j = c - J$
$m_{u_c[n] \to R_c}$	(5.28)

Table 5.1: A summary of all transmitted messages

number of measurements take so far.) As we will see in Section 5.1.3, the number of update iterations needed to converge is insensitive with respect to the number of cells C and can be considered as constant.

Distributed Implementation

This message passing algorithm can be easily implemented in a distributed fashion on our multi-robot system by partitioning the factor graph. As an example, the graph can be divided into four partitions, as shown in Figure 5.4. Each partition is assigned to one robot of our multi-robot system. Every robot is responsible for calculating all messages within its assigned partition on its on-board computers. Only messages crossing the boundary of the partitions have to be communicated to other robots. Note that therefore it is of interest to partition the graph in such a way that the boundaries are small. Smaller boundaries cause fewer messages to be send to other robots and require less communication. Further, we would like to remark that the partitioning of the graph and assignment of partitions does not correlate with the spatial position of the robots. A robot could be responsible for a partition that belongs to a spatial region of the environment while it is currently not physically in this region. In these cases also measurements taken in a certain region have to be communicated to the robot responsible for this particular region.

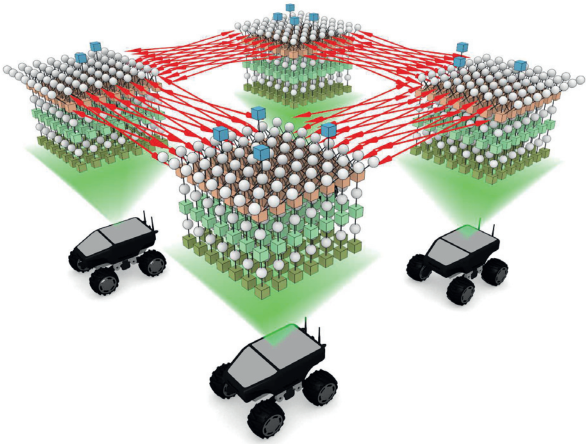


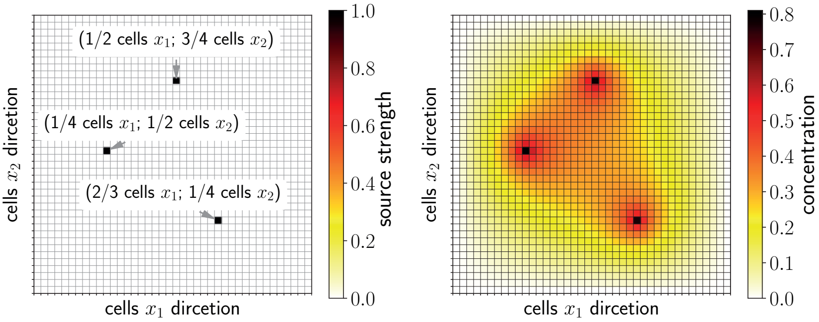
Figure 5.4: The figure illustrates how the factor graph can be divided into four parts. Each part is assigned to one robot. The robot calculates all messages within its part on its on-board computer. Only messages along common borders (red arrows) have to be exchanged between robots.

5.1.3 Performance Evaluation

In order to analyze the performance of the proposed inference algorithm, we set up a numerical simulation. The purpose of the simulation is to evaluate how well the algorithm is able to estimate the concentration and source distribution based on measurements. In addition, this section analyzes the convergence of the proposed algorithm. At this stage the exploration strategy — i.e. where the measurements are take — is out of the scope. Therefore, the measurements are take and randomly chosen locations.

For the evaluation we consider the static case of the linear diffusion equation (3.41) gained from FDM. We consider a quadratic domain with quadratic cells as shown in Figure 5.5. For simplicity the edge length ρ of a cell and the diffusion coefficient are set to 1. The size of the domain and therefore the number of cells is varied in order to analyze the effect of the discretization. We place three sources with strength 1 in the domain by designing a ground truth source distribution $\vec{u}_{gt}$. Figure 5.5 shows the ground truth concentration field as well as the ground truth sources distribution (sources correspond to the back squares). The ground truth concentration field $\vec{f}_{gt}$ can be gained by solving:

$$\vec{f}_{gt} = -\begin{bmatrix} \mathbf{D} \\ \mathbf{B} \end{bmatrix}^{-1} \begin{bmatrix} \vec{u}_{gt} \\ \vec{0} \end{bmatrix} \tag{5.29}$$



(a)

(b)

Figure 5.5: The performance evaluation uses a simulation setup as depicted in this figure. In the simulations different resolutions of the grid are analyzed. Here the case of 40x40 cells is shown. The three sources are always place at the same position in relation to the number of cells, e.g. at 1/4 of the grid's width and 1/2 of the grid's height. The three back cells in (a) illustrates the position of the three sources, while the plot in (b) shows the simulated gas concentration.

Here the equation system arises from the combination of the linear diffusion equation (3.41), **D**, and the Dirichlet boundary condition (3.14), **B**. Note that this forward problem is well posed since the boundary condition and source distribution are given.

Measurements are placed randomly in the domain where the actual simulated measured values are taken from the ground truth concentration $\vec{f}_{gt}$. Measurements are taken in a certain percentage of cells p_{msmt}. E.g. we may take measurements in $p_{msmt} = 70\%$ of the cells which means that 30% are not measured. Please note that in general 100% of the cells has to be measured in order to get a well posed inverse problem that can be solved without regularization. Loosely speaking, if we only use 70%, it implies that the algorithms has to "guess" 30% of the information.

Let's have a look at the convergence of the proposed message passing algorithm. The algorithm is implemented according to the previous sections where we have chosen $\tau_s = 10^3$ and $\tau_m = 10^5$. (For a detailed explanation and study of reasonable parameters see Section 7.1) Further, numerically it is not possible to set the precision in the update (5.28) to infinity. Thus, we set it to a very high number, i.e. 10^9. In each iteration of the algorithm all messages of the factor graph are calculated once in random order. That the algorithm in fact converges is shown in Figure 5.6 for different sizes of the domain and averaged over 100 runs. The error is calculated as the NMSE of the estimated concen-

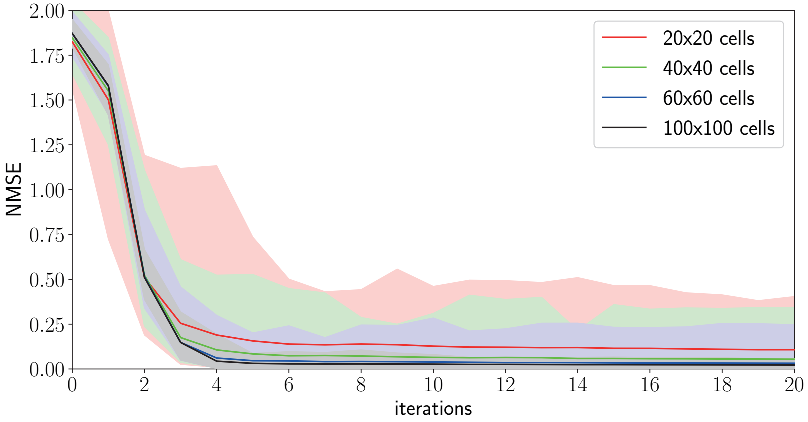
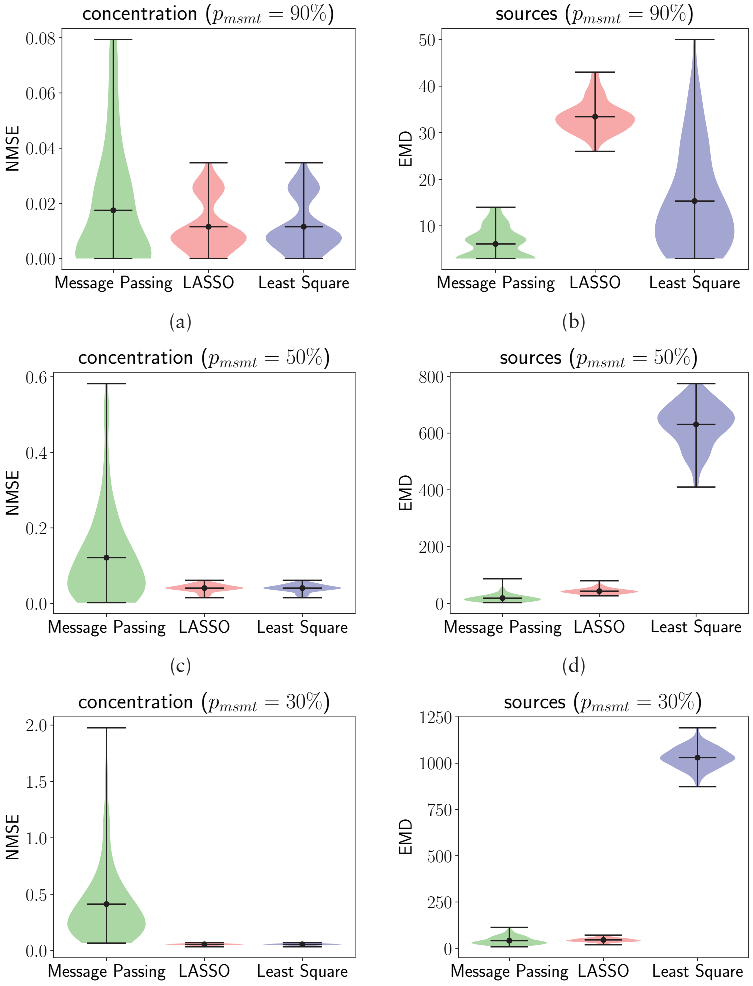


Figure 5.6: The curves show the convergence of the NMSE with respect to the number of message passing iterations. During one iteration all messages are calculated once. The number of messages to be calculated depends on the number of cells. The curves are averaged over 100 runs. The light colored areas are bounded by the maximum and minimum out of these 100 runs

tration $\vec{f}$ compared to $\vec{f}_{gt}$. As can be seen from the curves, the error of the estimation converges to small values. It is not a surprise that it does not reach zero, since “30% of the information is missing”. Further, it has to be remarked that by a higher resolution the finally achieve NMSE is better. Surprisingly, the algorithm converges for all different domain sizes within the same number of iterations (approximately 8). Of course the number of messages to be calculated in an iteration is higher for a lager domain size. However, this number and the computational time scale linearly with the number of cells. Further, it has to be remarked that by a higher resolution the finally achieve NMSE is better.

Now let’s analyze the quality of the estimate provided by the algorithm. To this end we compare the error of the estimate with other state-of-the-art solvers. Namely, we are using a least square solver and a LASSO solver which we already applied in Section 3.6. For the least square solver we use the pseudo-inverse function of the numpy [2] library. Further, we use the LASSO implementation of the scikit-learn [4] library ($\alpha = 0.4$). We compare the solvers by means of the NMSE of the estimated concentration. Additionally, we evaluate the EMD between the ground truth source distribution and the estimated source distribution. The EMD is especially suitable for measuring the error of sparse signals (see also Appendix C). Figure 5.7 shows the error depending on the number of measurements for a fixed resolution of 40x40 cells. The sources were placed as shown in Figure 5.5. Let’s first have a look at the case where



(a) (b)

(c) (d)

(e) (f)

Figure 5.7: The plots compare the quality of the estimates of concentration (first column) and source distribution (second column) depending on the number of measurements. The violin plots depict the probability density of the errors (NMSE and EMD) based on 100 simulation runs. The proposed message passing algorithm is compared to a least square and a LASSO solver. For the first row, 90% of the grid cells were measured, in the second row 50% and in the last row only 30%.

many measurements are available ($p_{msmt} = 90\%$). There the performance for estimating the concentration distribution is very similar for all three algorithms. Namely, the NMSE is below 0.1 for all estimates and the averages is close to 0.02 for all three. However, the performance for estimating the sparse source distribution differs. Message passing with SBL achieves a lower EMD compared to the LASSO and a lower variance in the EMD compared to the least square solver. For cases with fewer measurements ($p_{msmt} = 50\%$ and $p_{msmt} = 30\%$), the situation changes. The performance of the message passing algorithm gets worse for estimating the concentration. Nevertheless, the performance of estimating the sparse source distribution — measured by the EMD — remains better compared to the LASSO and much better compared to the least square solver. As a further note: for fewer measurements the performance of the LASSO gets better compared to least square for estimating the source distribution.

To sum up, for our purpose of source estimation the proposed message passing algorithm performs very well, especially in cases were only few measurements are available. The reason for that is the sparsity inducing prior implemented with SBL. This prior helps the algorithm to "guess" the right solution in case the measurements do not provide enough information.

5.2 Approximate Solution of the Non-Linear Model

In the previous section we developed a message passing algorithm to calculate mean and variances of the posterior (4.1) (more precisely the mean and variances of marginal distributions of the posterior). Unfortunately, we can only apply this algorithm in case of a linear model, i.e. as long as the residuum (4.4) is linear in the unknown concentration and source distribution. As explained in section 3.3, the advection-diffusion PDE with unknown air flow is non-linear. The reason for the non-linearity is the bi-linear term in equation 3.38 where the air flow $\vec{v}_1[n]$ is multiplied by the concentration $\vec{f}[n]$. Due to the non-linearity and due to the form of the used hierarchical prior, calculating mean and variances of the posterior is generally harder. Therefore, we make use of variational inference techniques to approximate the posterior. This approach was first published in [188].

To simplify the following notation, during this section we introduce the auxiliary variables $\vec{w}[n] = [\vec{f}[n]^T, \vec{v}_1[n]^T, \vec{v}_2[n]^T, \vec{f}[n-1]^T]^T$ that contains the concentration at the current and the previous time stamp as well as the wind field. Likewise, we introduce the auxiliary variables $\vec{\theta}[n] = [\vec{w}^T, \vec{u}[n]^T]^T$ that is an extension of $\vec{w}[n]$ by the source distribution. Further, we define $\mathcal{Y}$ as the set $\{\vec{y}[0] \ , \dots , \ \vec{y}[n]\}$ of all measurements taken until time stamp n. We will from now on not refer explicitly to the dependency of variables on the time index n to further simplify notation. Now, consider the evidence of the model $p(\mathcal{Y})$:

$$p(\mathcal{Y}) = \int p(\mathcal{Y}, \vec{\theta}, \vec{\gamma}) d\vec{\theta} d\vec{\gamma}. \tag{5.30}$$

The key idea of a variational Bayesian inference is to introduce a proxy distribution $q(\vec{\theta}, \vec{\gamma})$ to approximate the posterior of interest as follows [32] :

$$\begin{aligned} \log p(\mathcal{Y}) &= \log \int q(\vec{\theta}, \vec{\gamma}) \frac{p(\mathcal{Y}, \vec{\theta}, \vec{\gamma})}{q(\vec{\theta}, \vec{\gamma})} d\vec{\theta} d\vec{\gamma} \\ &\geqslant \int q(\vec{\theta}, \vec{\gamma}) \log \frac{p(\mathcal{Y}, \vec{\theta}, \vec{\gamma})}{q(\vec{\theta}, \vec{\gamma})} d\vec{\theta} d\vec{\gamma} = \mathcal{L}(q(\vec{\theta}, \vec{\gamma})). \end{aligned} \tag{5.31}$$

The difference between the true log marginal probability and the lower bound $\mathcal{L}(q(\vec{\theta}, \vec{\gamma}))$ is known to be Kullback-Leibler divergence from $q(\vec{\theta}, \vec{\gamma})$ and the posterior (4.1):

$$KL(q||p) = -\int q(\vec{\theta}, \vec{\gamma}) \log \frac{p(\vec{\theta}, \vec{\gamma}|\mathcal{Y})}{q(\vec{\theta}, \vec{\gamma})} d\vec{\theta} d\vec{\gamma}. \tag{5.32}$$

Since $\log p(\mathcal{Y})$ is a constant quantity for given observations, by maximizing the lower bound $\mathcal{L}(q(\vec{\theta}, \vec{\gamma}))$ we minimize the Kullback-Leibler divergence and $q(\vec{\theta}, \vec{\gamma})$ will become close to the desired posterior $p(\vec{\theta}, \vec{\gamma}|\mathcal{Y})$.

In order to solve the variational problem, i.e. maximizing $\mathcal{L}$ with respect to $q(\vec{\theta}, \vec{\gamma})$, we make use of the mean field approximation:

$$q(\vec{\theta}, \vec{\gamma}) = q_\theta(\vec{\theta}) \prod_{i}^{C} q_{\gamma_i}(\vec{\gamma}_i) \tag{5.33}$$

In SBL the approximation PDFs q_θ and q_{γ_j} are chosen from conjugate families. This choice ensures that during the updates the distributions keep their representation and stay in their family. Further, it ensures analytical tractability of the inference expression. Classically in SBL, the posterior would be a combination of a Normal distribution and a Gamma distribution, the choice we also make here. In particular, we choose q_θ and q_{γ_j}, $j = 1, \ldots, C$ as (see also [32]):

$$q_\theta(\vec{w}, \vec{u}) = N(\vec{\theta}|\vec{\mu}_\theta, \vec{\Sigma}_\theta) = N\left(\begin{bmatrix} \vec{w} \\ \vec{u} \end{bmatrix} \middle| \begin{bmatrix} \vec{\mu}_w \\ \vec{\mu}_u \end{bmatrix}, \begin{bmatrix} \vec{\Sigma}_{w,w} & \vec{\Sigma}_{w,u} \\ \vec{\Sigma}_{u,w} & \vec{\Sigma}_{u,u} \end{bmatrix} \right) \tag{5.34}$$

$$q_{\gamma_j}(\vec{\gamma}_j) = G(\vec{\gamma}_j|a_j, b_j),\ j = 1...C. \tag{5.35}$$

Reconsidering the dimension of the concatenated vectors $\vec{w}$ and $\vec{\theta}$, the covariance matrices $\vec{\Sigma}_\theta$ is in $\mathbb{R}^{5C\times5C}$ (respectively: $\vec{\Sigma}_{w,w} \in \mathbb{R}^{4C\times4C}$ and $\vec{\Sigma}_{u,u} \in \mathbb{R}^{C\times C}$).

Now, using (5.33), we iteratively maximize $\mathcal{L}$ by updating each factor of $q(\vec{\theta}, \vec{\gamma})$ as follows [32]:

$$q_\theta(\vec{w}, \vec{u}) \leftarrow \frac{1}{\alpha} \exp\left(\left\langle \log p(\mathcal{Y}, \vec{\theta}, \vec{\gamma}) \right\rangle_{q_{\gamma_i}|i=1...C} \right) \tag{5.36}$$

$$q_{\gamma_j}(\vec{\gamma}_j) \leftarrow \frac{1}{\beta}\exp\left(\left\langle \log p(\mathcal{Y}, \vec{\theta}, \vec{\gamma}) \right\rangle_{q_\theta, q_{\gamma_i}|i=1...C, i\neq j}\right), \; j = 1...C, \tag{5.37}$$

where $\langle a(x) \rangle_{b(x)}$ stands for an expectation of a function $a(x)$ with respect to the PDF $b(x)$, and α and β are normalization constants.

Let's start to apply the update rules to our problem. In our case we obtain

$$q_\theta(\vec{w}, \vec{u}) \leftarrow \frac{1}{\alpha}\exp\left(\langle \log(p(\mathcal{Y}|\vec{w})p(\vec{w}|\vec{u})p(\vec{u}|\vec{\gamma})p(\vec{\gamma}))\rangle_{G(\vec{\gamma_i}|a_i,b_i)|i=1...C}\right) \tag{5.38}$$

which requires evaluating

$$\begin{aligned}\hat{q}_\theta(\vec{w}, \vec{u}) = \exp\Bigg(\Bigg\langle &\log(p(\mathcal{Y}|\vec{w})p(\vec{w}|\vec{u})) - \sum_c \gamma_c \vec{u}_c^2 - \\ &- \sum_c \log\frac{1}{\vec{\gamma_c}}\Bigg\rangle_{G(\vec{\gamma_i}|a_i,b_i)|i=1...C}\Bigg)\end{aligned} \tag{5.39}$$

To compute the expectations we make use of the following facts

$$\begin{aligned}\langle \log(p(\mathcal{Y}|\vec{w})p(\vec{w}|\vec{u}))\rangle_{G(\vec{\gamma_i}|a_i,b_i)|i=1...C} &= \log(p(\mathcal{Y}|\vec{w})p(\vec{w}|\vec{u})) \\ \left\langle \sum_c \gamma_c \vec{u}_c^2 \right\rangle_{G(\vec{\gamma_i}|a_i,b_i)|i=1...C} &= \sum_c \frac{a_c}{b_c}(u_c)^2 \\ \left\langle \sum_c \log\frac{1}{\vec{\gamma_c}} \right\rangle_{G(\vec{\gamma_i}|a_i,b_i)|i=1...C} &= const. \text{ with respect to } \vec{w} \text{ and } \vec{u}.\end{aligned}$$

This results in

$$\hat{q}_\theta(\vec{w}, \vec{u}) \propto p(\mathcal{Y}|\vec{w})p(\vec{w}|\vec{u}) \prod_c N\left(\vec{u}_c|0, \left(\frac{a_c}{b_c}\right)^{-1}\right) \tag{5.40}$$

Unfortunately because of the non-linearity in (3.38), $\hat{q}_\theta(\vec{w}, \vec{u})$ is not a normal distribution. The factor $q_\theta(\vec{w}, \vec{u})$ in (5.34) is, however constrained to be normal. Therefore, we minimize the Kullback-Leibler divergence from $q_\theta(\vec{w}, \vec{u})$ to $\hat{q}_\theta(\vec{w}, \vec{u})$. To this end, we need the moments of $q_\theta(\vec{w}, \vec{u})$. These can be achieved by setting the mean of $q_\theta(\vec{w}, \vec{u})$ to the maximizer of $\hat{q}_\theta(\vec{w}, \vec{u})$ with respect $\vec{\theta}$, and covariance of $q_\theta(\vec{w}, \vec{u})$ to the curvature of $\hat{q}_\theta(\vec{w}, \vec{u})$ around the maximizing value. For this task we use a numerical solver. In particular, we use the Newton-CG optimization implementation in scipy [5]. Further, from a numerical perspective it makes sense to consider the optimization in the log domain as follows.

$$\vec{\mu}_\theta = \arg\max_{\vec{\theta}} \; \log \hat{q}_\theta(\vec{\theta}). \tag{5.41}$$

To estimate the covariance matrix $\vec{\Sigma}_\theta$ we compute the inverse of the Hessian matrix evaluated at $\vec{\mu}_\theta$:

$$\vec{\Sigma}_\theta = \left[-\frac{d^2 \log \hat{q}_\theta(\vec{\theta})}{d\vec{\theta}_i d\vec{\theta}_j} |_{\vec{\mu}_\theta} \right]^{-1} \tag{5.42}$$

Now, we proceed similarly with factors q_{γ_j}, $j = 1, \dots, C$. We compute

$$\begin{aligned} q_{\gamma_j}(\vec{\gamma}_j) \leftarrow \frac{1}{\beta}\exp\Bigg(\Bigg\langle & \log(p(\mathcal{Y}|\vec{w})p(\vec{w}|\vec{u})) - \sum_c \gamma_c \vec{u}_c^2 - \\ & - \sum_c \log\frac{1}{\vec{\gamma_c}} \Bigg\rangle_{N(\vec{\theta}|\vec{\mu}_\theta,\vec{\Sigma}_\theta),G(\vec{\gamma_i}|a_i,b_i)|i=1 \dots C,i\neq j}\Bigg), \end{aligned} \tag{5.43}$$

with

$$\begin{aligned} \langle\log(p(\mathcal{Y}|\vec{w})p(\vec{w}|\vec{u}))\rangle_{N(\vec{\theta}|\vec{\mu}_\theta,\vec{\Sigma}_\theta),G(\vec{\gamma_i}|a_i,b_i)|i=1 \dots C,i\neq j} &= \text{const. with respect to } \gamma_j \\ \left\langle \sum_c \gamma_c \vec{u}_c^2 \right\rangle_{N(\vec{\theta}|\vec{\mu}_\theta,\vec{\Sigma}_\theta),G(\vec{\gamma_i}|a_i,b_i)|i=1 \dots C,i\neq j} &= (\vec{\Sigma}_{u_j,u_j} + \vec{\mu}_{u_j}^2)\gamma_j + \text{const.} \\ \left\langle \sum_c \log\frac{1}{\vec{\gamma_c}} \right\rangle_{N(\vec{\theta}|\vec{\mu}_\theta,\vec{\Sigma}_\theta),G(\vec{\gamma_i}|a_i,b_i)|i=1 \dots C,i\neq j} &= \text{const. with respect to } \gamma_j. \end{aligned}$$

This results in:

$$\begin{aligned} q_{\gamma_j}(\vec{\gamma}_j) \leftarrow \frac{1}{\beta}\exp\left(-(\vec{\Sigma}_{u_j,u_j} + \vec{\mu}_{u_j}^2)\vec{\gamma}_j\right) &\propto G(\gamma|1,(\vec{\Sigma}_{u_j,u_j} + \vec{\mu}_{u_j}^2)) \\ \text{i.e. } a_j = 1 \text{ and } b_j &\leftarrow \vec{\Sigma}_{u_j,u_j} + \vec{\mu}_{u_j}^2. \end{aligned} \tag{5.44}$$

Ultimately, we can combine the two update rules in the following algorithm:

1. initialize $\hat{\gamma}_i$; $i = 1 \dots C$ with a low value
2. compute the mean and variance of $q_\theta(\vec{w}, \vec{u})$ using (5.41) and (5.42).
3. update $q_{\gamma_j}(\gamma_i)$, $j = 1, \dots, C$ from (5.44) .

By repeating step 2 and 3 iteratively, we get our approximated solution of the posterior $q_\theta(\vec{w}, \vec{u})$ and $q_{\gamma_j}(\vec{\gamma_j})$, $j = 1, \dots, C$. The computational complexity of this algorithms is dominated by the inversion of the Hessian in equation (5.42) for calculating the covariance matrix. In the worst case, when the sparse structure of the Hessian matrix is not exploited, the complexity is therefore $\mathcal{O}(C^3)$, where C is the number of cells or nodes used to discretize the environment.

At this point we would like to remark that one would get the same update rules following the Expectation Maximization (EM) algorithm. For more details we refer to the appendix in [174].

5.3 Summary

In this chapter we provided two algorithms to calculate interesting features of the posterior PDF introduced in chapter 4. Namely, we are interested in the mean and variance of the posterior PDF. Since the mean is the most likely combination of concentration, source and wind distribution, it is actually the solution to our inverse problem. The variance will get important in the next chapter for deriving the exploration strategy, since it can be used as a gauge of information. The first presented algorithm can be applied to the linear dispersion model. This novel algorithm is based on message passing on factor graph and therefore extremely suitable for a distributed implementation. It shows a better performance when estimating the sparse source distribution compared to LASSO or a least square solvers. The second algorithm makes use of variational inference in order to cope with the non-linearity of the dispersion model. For the second algorithm, so far only a centralized version is presented, where a distributed implementation is part of future work.

Chapter 6
Exploration and Source Localization Strategy

In the previous chapters we designed a PDE based gas dispersion model in a probabilistic framework that allows to infer the parameters of gas sources based on concentration measurements. In this chapter we are going to make use of this probabilistic model to design our Domain-knowledge Assisted Robotic Exploration and Source-localization (DARES) strategy. DARES is used to coordinate multiple robots and to guide them to informative new measurement locations. The main objective of DARES is to localize sources efficiently. That means: as fast as possible and with as few measurements as possible. Besides, we have to coordinate the multi-robot system in a way to avoid inter-robot collisions. Before presenting our approach, we summarize the state-of-the-art of gas source localization strategies.

6.1 Related Work

The use of robots for gas source localization is an active research field since the 90s of the last century [87]. Here we focus our short survey on gas source localization approaches according to the topic of this thesis. Nevertheless, there exists a certain overlap to other closely related research fields like gas distribution mapping or plume tracking. For an overview of different approaches and terminology we refer the reader to the review papers [87], [86], [196], [103], [201]. Further, we focus on adaptive - sometimes called reactive - source localization strategies. Adaptive strategies react in real time to measurements and adapt the robot's path accordingly, in contrast to proactive (or predefined) strategies. Proactive strategies follow a fix predefined path often with the objective to cover the whole environment [98]. In this thesis we only refer to one predefined sweeping trajectory in Section 7.3 as a benchmark to compare our proposed strategy.

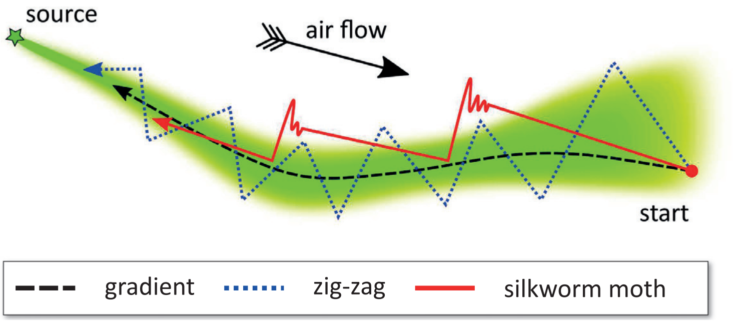


Figure 6.1: The Figure illustrates three common bio-inspired gas source localization strategies (adapted from on [153]). The gas plume and gas concentration are depicted from yellow (low concentration) to green (high concentration). The “gradient” approach follows the concentration gradient towards the source. The other two strategies follow the air flow in upwind direction and adapt their trajectories as soon as they loose the plume. The figure shows idealized rather than typically observed trajectories.

6.1.1 Bio-Inspired Gas Source Localization Strategies

Searching for chemical sources is also a challenging task for a lot of animals or bacteria. Popular examples are moths looking for mates, lobsters searching for food, escherichia coli bacteria seeking for nutrients or dung beetles localizing feces [87]. Therefore, in the field of MRO scientists try to mimic or copy such behavior to mobile robots.

One category of bio-inspired strategies are gradient based approaches [130], [115], [133]. Often this kind of approaches are referred to as chemotaxis [196], [82]. Gradient based approaches for gas source localization assume that the gas concentration rises monotonously when approaching the source. By following the spatial gradient of the gas concentration the robot is supposed to ends up at the source location. An example of this approach are the Braitenberg vehicles [34] in [99]. Unfortunately, gas dispersion is typically dominated by turbulence and air flow causing a chaotic, patchy and intermittent distribution [113]. Therefore, the concentration gradient does not necessarily lead towards the gas source [133]. Furthermore, since the robot tends to get stuck at the location of the source (location of the highest concentration), a purely gradient based approach is not suitable for searching for multiple sources.

In addition to pure concentration measurements, anemotaxis also considers observations of the air/fluid flow, e.g. in [153], [69], [82], [162]. Since air flow is the main mechanism transporting the gas plume from the source to the robot, anemotaxis instructs the robot to move in upwind direction as soon as the robot detected gas. This strategy is expected to lead the robot to the source similar to gradient-based chemotaxis.

The zig-zag or dung beetle algorithm [83],[153],[130] is a popular bio-inspired algorithm making use of chemotaxis and anemotaxis. The algorithm follows the plume in upwind direction in a zig-zag pattern and the robot is turning back to the plume every time it is reaching the plume's border. Another algorithm that uses chemo- and anemotaxis mimics the silkworm moth's behavior [100], [153],[130]. If the silkworm moth algorithm is triggered by a high concentration measurement, the robot moves in a certain pattern based on three steps. First the robot moves in upwind direction for a fixed amount of time. Then it starts to move side to side in a zig-zag fashion, followed by a circular movement in order to find the plume again. Whenever a high concentration is detected, the pattern re-starts with moving in upwind direction. This silkworm moth algorithm is similar to the surge-cast algorithm proposed in [106]. Figure 6.1 illustrates the trajectories of the zig-zag, silkworm moth and the gradient based algorithm.

6.1.2 Infotaxis and Probabilistic Source Localization

In the last decades, more sophisticated gas source localization strategies have emerged, often no longer bio-inspired. Increasingly many researchers nowadays approach gas source localization in a probabilistic framework [118],[11],[56],[152]. This is reasonable in order to account for the often chaotic and random behavior of gas dispersion caused by e.g. turbulence [113]. Probabilistic approaches lend themselves naturally to exploration strategies based on information theory. This class of strategies has been labeled as Infotaxis [180]. They aim at maximizing the information on the gas sources. Maximizing the information gain is often equivalent to an entropy reduction [126]. For instance in [25] the objective of a path planner is to reduce the entropy of a sensor- and gas dispersion model using the Fischer information matrix. Comparably, in [11] an information-theoretic sensor-path-planning algorithm is developed to reduce the uncertainty about the source emission rate. As an alternative to reducing the entropy, in [56] the proposed robot path planner maximizes the likelihood of detecting a source based on a probabilistic source map.

Let us remark that infotactic approaches in general need a mathematical model of the gas dispersion process or at least a sensor model. This model is required to define and calculate the information gain, entropy or Fischer information matrix. Besides simple Gaussian Plume models as in e.g. [152], the advection-diffusion equation was used quite often [126],[70],[118] as a dispersion model. The strong dependence on an accurate dispersion model is a major limitation of Infotaxis. As soon as the model does not represent the reality accurately, the performance of infotactic strategies drops [149].

6.1.3 Optimal Experimental Design and Optimal Sensor Placement

Finding an exploration strategy is closely related to an optimal sensor placement problem and optimal experimental design [139],[178]. These methods from the field of control theory try to find optimal sensor locations to observe a distributed parameter system, similar to our concentration distribution governed by a PDE [89]. For example, in [52] a controller for a multi-agent system was designed considering the performance of a state estimator for a spatio-temporal process represented by a PDE. Similar, in [51] the authors make use of a Luenberger observer for an advection-diffusion PDE modeling gaseous release. They design a controller for multiple UAVs coupling the dynamic constraints of the UAVs and the observer performance. Observer design techniques are also explicitly applied for plume tracking. In [95] a non-linear observer is designed to track a plume with a multi-robot system. And in [112] the formation of a robotic swarm is optimized by a coverage criterion to track a plume modeled as a Gaussian Plume. Likewise, [163] proposes a source localization strategy based on wind and concentration measurements, where a flock of robots positions itself relative to a gas plume in order to track it in upwind direction. In general, such approaches are very similar to infotactic source localization strategies, since they are often making use of the Fischer information matrix [179]. In the field of optimal experimental design different properties of the Fischer information matrix are optimized to determine optimal sensor locations. The most popular ones are A-optimality and D-optimality criteria. The A-optimality criterion optimizes the trace of the inverse information matrix, which can be interpreted as the optimization of the average parameter's variance. The D-optimality optimizes the determinant of the information matrix, which corresponds to the optimization of the volume of the parameter's uncertainty ellipsoid [139]. For example, D-optimality is applied in [176] to a 2D diffusion PDE in order to plan optimal trajectories for mobile sensors. In [12] the A-optimality in a Bayesian context is investigated to place sensors in order to estimate the initial condition of a PDE.

6.2 Domain-knowledge Assisted Robotic Exploration and Source-localization (DARES) strategy

The key idea of our Domain-knowledge Assisted Robotic Exploration and Source-localization (DARES) strategy is to direct robots to regions, where our knowledge about the explored process is currently uncertain. We assume that measurements in these regions are more suitable to estimate the source distribution (i.e. to localize the sources) compared to measurements in regions where we already know the process very well. Therefore, our proposed DARES approach can be classified as infotaxis. Like infotactic strategies in general, our approach requires a mechanism for quantifying the information we have about the process itself and about process parameters. In particular, in our context if the uncertainty about the gas concentration value or source strength at a location is high, this location is a good candidate for the next measurement location. In the following we present how the uncertainty can be quantified based on the developed probabilistic model. The probabilistic model introduced in Chapter 4 provides the posterior probability of all combinations of discretized concentration values, source strengths and wind velocities. One may call this a global point of view on the process. However, for DARES, we want to quantify the uncertainty at individual locations and we want to compare these locations to find the most uncertain location. To this end, we calculate marginal PDFs for each location based on the posterior (4.1). Note that in our discretized environment a location corresponds either to a grid cell c in case of FDM or a mesh node c in case of FEM. In the following we are using the term location as a substitution for either a cell or a node. In general the marginal PDF of the concentration or source strength at a location could be calculated by integrating over all other variables and parameters. E.g. for the concentration value $f_c[n]$ the marginal is defined as:

$$\begin{aligned}p(f_c[n]|\vec{y}[0],...,\vec{y}[n]) \propto \int ... \int & p(\vec{f}[n],\vec{u}[n],\vec{v}_1[n],\vec{v}_2[n],\vec{f}[n-1]|\vec{y}[0],...,\vec{y}[n]) \\ & df_0[n]...df_{c-1}[n]df_{c+1}[n]df_C[n]du_0[n]...du_C[n] \\ & dv_{1,0}[n]...dv_{1,C}[n]dv_{2,0}[n]...dv_{2,C}[n]\end{aligned} \tag{6.1}$$

Actually, it is not necessary to solve these computationally expensive integrals explicitly. Based on the distributed message passing algorithm presented in Section 5.1.2 we compute the marginal distribution of the source strength $u_c[n]$ and gas concentration $f_u[n]$ for free. In particular, the outgoing messages of variable nodes $u_c[n]$ and $f_u[n]$ are the marginal PDFs we are looking for. It has been shown in Section 5.1.2 that the marginal distributions of both gas concentration and source intensity are Gaussian PDFs. As such, the second order moments of these variables can be used as a gauge of the information content at the location c (grid cell or mesh node). Note that for a Gaussian random variable the entropy is related to the square root of the variable's variance. Thus, the variance of either the source or concentration marginal PDF can be used to quantify the uncertainty; locations with a higher variance are more uncertain and therefore are more "interesting" as potential location for a new measurement. In this context interesting means that a measurement at this location will help to reduce the error in the parameter estimation more compared to other locations. Let us remark that an accurate estimation of the source distribution actually implies a successful localization of the sources.

Unfortunately, as has been shown in Chapter 5 for a non-linear model the message passing algorithm is not applicable. However, the solution presented in

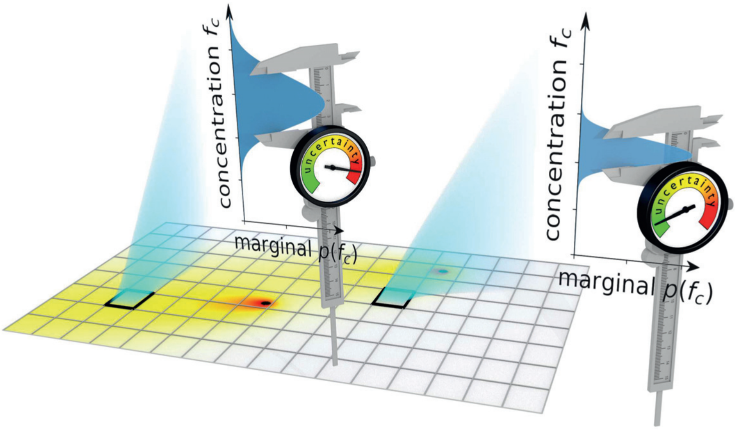


Figure 6.2: The figure illustrates the uncertainty quantification used for the DARES strategy. For each grid cell the marginal gas concentration PDF is available. If the variance of the marginal PDF is high, the cell is considered to be more uncertain compared to a cell with lower variance.

Section 5.2 provides an approximate Gaussian PDF for the multivariate posterior. More precisely, it provides the covariance matrix of the posterior based on equation (5.42). As it is commonly known the marginal distribution of as multivariate Gaussian is a Gaussian again, where the variance is the corresponding diagonal elements of the covariance matrix. As such, also for the non-linear case approximations of the marginal PDFs are easy to obtain.

Based on the variances of the marginal PDFs of the concentration for each region, we can create an uncertainty map for the whole environment. A high variance in the concentration marginal PDFs corresponds to a high uncertainty in the map. While in general the uncertainty map could be also generated based on the variance of the marginal PDFs of the sources, the concentration marginals seem more intuitive. This is because we are able to directly measure the concentration, and therefore directly reduce the uncertainty in the concentration by new measurement. Even though concentration and sources are tightly coupled by the dispersion model, the source are not directly measurable and a concentration measurement would not directly reduce the uncertainty in the sources.

To sum up: the variance of the marginal PDFs of the concentration value at a certain location is used to propose potential locations of interest for making new concentration measurements. All locations are rated according to the inverse variance, i.e., precision. A pre-defined number of locations with the lowest precision are selected as a proposal for new measurement locations. The robots of the multi-robot system select from this set of measurement locations. While this approach seems reasonable from the information theoretical point of view, a simulation and experimental evaluation will be provided in the next chapter.

6.3 Multi-Robot Coordination

As described in the previous section, our DARES approach generate a list of of new measurement locations for the robots. This section explains how the multiple robots choose from this list and how they are coordinated so that they do not collide on their way to their next measurements location. Since the gas dispersion process is considered as a two dimensional phenomenon, this section assumes ground based robots which operate only in a two dimensional plane. We assume that the robots are able to localize themselves perfectly which means with an accuracy much higher than the spatial discretization. Further, the robots have the capability to move to a desired way-point and avoid static obstacles marked in an obstacle map. To address the path planning problem we make use of state-of-the-art solutions and implementations.

The multi-robot system is following the procedure shown in Figure 6.3. As explained in the previous section, based on the probabilistic gas dispersion model we can generate a set of locations where new measurements are most helpful to improve our estimation of the gas dispersion process and thus help to localize the sources. This set of candidate locations can either be generated by a single processing unit or in a distributed fashion (see Section 5.1.2). The robots choose their next way-point from this set and move there. However, choosing from the set of locations and reaching the chosen location is not a trivial problem. The multi-robot system has to fulfill additional constraints.

Most importantly, the robots have to make sure that they do not collide when moving toward their next goal. In addition, two robots should avoid approaching the same location for reasons of efficiency. In our experimental evaluation we propose two different mechanisms to avoid collisions between robots and coordinate the robots. These mechanism will be explained a bit further down.

At this point let us have a closer look at the system presented in the flow chart in Figure 6.3. As can be seen, the robotic navigation and the update of the probabilistic model are actually two separate processing loops. These loops run independently of each other and asynchronously. In other words, the navigation loop of the robots is not synchronized with the model update loop. It just considers the last generated set of interesting locations.Further the navigation loop does not run with a constant frequency, since the time to reach a new goal always differs depending on the distance to the new location. We would like to stress that the navigation loop of each robot is also not synchronized with other robots. Thus, no robot has to wait for results of the other robots. Whenever a measurement becomes available for a time stamp,

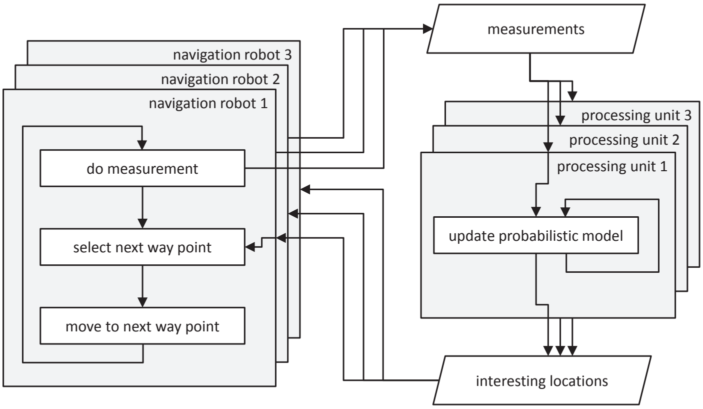


Figure 6.3: Flow chart depicting the procedure of the DARES approach: It mainly consists of two independent, unsynchronized loops, which either take care of the robot's navigation or processing and generating of new set of way point proposals.

it is directly inserted into the dispersion model. As can be seen, the two loops are only connected by the exchange of the proposed measurement location and taken measurements. The update rate of the model and the periodic generation of interesting locations are fix. It is actually the time discretization Δt of the dynamic model (see Section 3.3.3) or an arbitrary time in the order of one to several seconds in our setting for the static case. It is important that update rate is fast enough so that the set of locations proposed to the robots never gets empty. In other words, the proposed set of locations must be generated faster than the time required by the robots to visit all of them. The robot's navigation loop is implemented in two different ways to account for collisions with other robots as described below.

6.3.1 Collision Avoidance: Rescheduling

Based on the probabilistic gas dispersion model the set of interesting location is periodically updated. The locations are further rated according to their informativeness, i.e. uncertainty. This ordered set of locations is then sent to all robots. Whenever a robot is in need of a new way-point, it selects the point with the highest uncertainty from the set. The robot moves to the selected way-point while periodically communicating its current location and goal to all other robots. The other robots mark the selected goal as uninteresting for themselves. When the robot reaches the goal, it takes a measurement. This measurement is either sent to all other robots (in case of a distributed algorithm) or to a central processing unit (in case of a central implementation). With the new measurement the probabilistic model is updated and a new set of proposals for measurement locations is created. The robot does not wait for this update, since the current set of locations still contains enough interesting locations. It will thus select a new location from the current set. This behavior is the same for all robots.

To avoid collisions the robots periodically check their distance to all other robots, while they are moving. If the distance is lower than a safety distance, both robots stop and select another goal. In this situation the robots do not choose from the hole set. Instead the goal is only chosen from those locations in the set so that going there will increase the critical distance again. If no such location is available, the robot has to wait either for an update of the set of interesting measurement locations or it has to wait for the other robot within the safety distance to clear the way. This algorithm is illustrated in Figure 6.4.

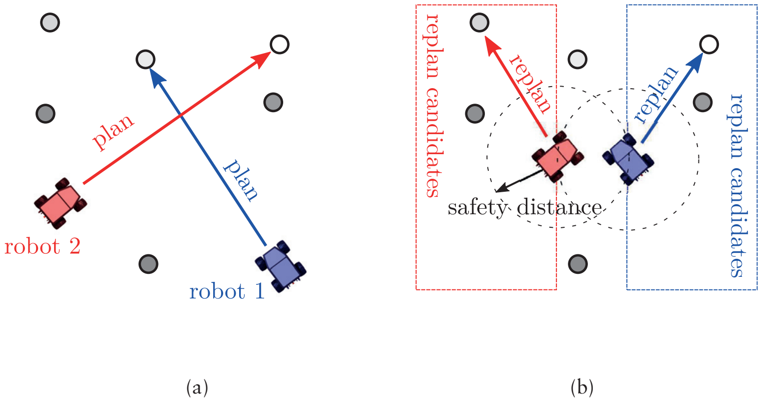


Figure 6.4: Illustration of the collisions avoidance mechanism based on rescheduling: In (a) robot 1 and robot 2 select an interesting location as their next goal (lighter colors correspond to more interesting locations). They follow a straight line trajectory towards their goal. As soon as both robots get too close to each other (below safety distance) in (b), both robots have re-planed and selected a new goal that will increase their mutual distance.

6.3.2 Collision Avoidance: Evading

Analogously to the previous collision avoidance mechanism, the probabilistic model generates the set of interesting locations periodically and the rated locations are communicated to all robots. Again each robot selects a location from the set and moves there periodically checking the distance to other robots. For that the robots sent their current location and goal to all other robots periodically. If the distance between two robots gets lower than the safety distance and a collision is impending, both robots stop. For this collision avoidance mechanisms priorities are assigned to each robot. The robot with the lower priority keeps standing still. The robot with the higher priority adds the other robot to its static obstacle map and continues toward its goal. In this way the path planner will avoid the other static robot considering it as an obstacle. As soon as the robots are further away than the safety distance plus a hysteresis, the static robot is removed from the obstacle map and starts to move again. Unfortunately, this approach may cause deadlocks. For example, if robots with lower priority are blocking the goal of a robot with a high priority. In this case the path planner of the robot will not find a way to the goal and the robot has to keep where it is. The other robots cannot clear a way because they have no permission to drive since they are to close to a robot with a high priority. To solve such constellations the low priority robots observe continuously the movement of robots with higher priority in their range. If the robot with higher priority does not move (since it is blocked) for a certain time, the robot with low priority re-plans its path to the goal considering the robot with higher priority in its static obstacle map. It will follow this path very slowly as long as the other robot is within the critical safety distance. As soon as it detects that a robot with higher priority within the safety distance starts to move, it will immediately stop again.

When a robot reaches its goal it takes a measurement and send the measurement to the processing unit that updates the probabilistic model. Afterward, it can select a new location from the current set of proposed interesting locations. This algorithm is illustrated in Figure 6.5. Further, Appendix B presents a detailed flow chart of the collision avoidance algorithm.

The two collision avoidance mechanisms differ in two important points. Evading makes sure that the original chosen measurement location will be reached by the robots. In contrast, when rescheduling the robots choose other location in the re-planning step. Therefore, the actually most interesting location may not be measured. However, rescheduling has a significant advantage. While in case of the evading robots sometimes have to wait until others clear the way, rescheduling makes sure that robots are always moving. Thus, the rescheduling mechanism is preferable if the environment is small compared to the number and size of robots. In a small environment the evading mechanism can be more inefficient, since robots may often block each other and only the robots with high priority are allowed to move. On the other hand, evading has an edge in larger environments, where robots rarely get close to each other.

It is important to note that both mechanisms are not optimal from a global point of view. As can be seen in constellations shown in Figure 6.4 and 6.5, the goals could have been assigned in a way that avoids the possibility of a colli-

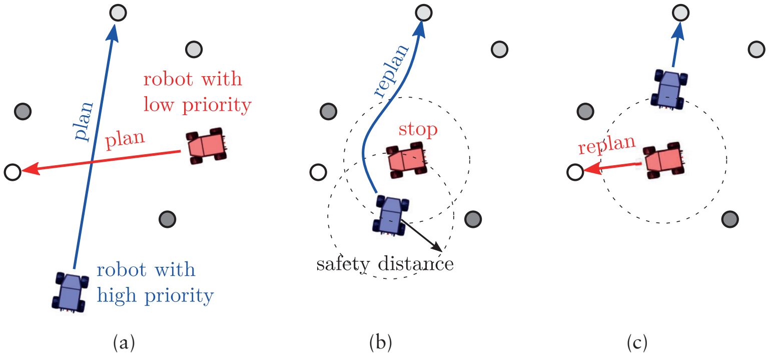


Figure 6.5: The figure illustrates the collisions avoidance mechanism based on evading. In (a) robot 1 and robot 2 select an interesting location as their next goal (lighter color more interesting) and follow trajectories towards their goals. As soon as both robots get to close to each other (below safety distance) in (b), both robots stop. The robot with higher priority (here blue), re-plans a new trajectory towards the its goal while the second robot (here red) is considered as a static obstacle. In (c), when the distance between the robots is large enough again, also the second robot with lower priority (here red) re-plans a trajectory to its goal.

sion. (The robots would just need to swap their goals.) However, the presented collision avoidance mechanisms are based on simple rules with little communication overhead. They scale nicely with the number of robots and they are robust. More sophisticated approaches can be considered in future work.

6.4 Summary

Based on the gas dispersion model which encodes our domain knowledge, in this chapter we introduced the DARES strategy for a multi-robot system. The DARES strategy guides the robots to informative locations for taking new measurements. In this chapter we presented how to quantify uncertainty of different locations in the environment based on the probabilistic model presented in Chapter 4 and the solvers presented in Chapter 5. This chapter further presented the system setup to coordinate multiple robots which cooperatively explore the environment by taking measurements. At this point we have fused all ingredients into the DARES strategy developed in this thesis to localize gas sources. Now, we are in the position to evaluate the proposed approach in simulations and experiments.

Chapter 7
Evaluation

7.1 Simulations

Simulations are essential to evaluate gas source localization algorithms. In contrast to real-world experiments, they permit full control of environmental parameter, like the position of the source, wind, etc. Furthermore, simulations are reproducible and facilitate to carry out a statistically significant number of experiments for evaluation. Also while ground truth data may be available for source locations, real-world experiments lack of ground truth data of the actual gas concentration and its distribution.

Even though this work mainly focuses on gas source localization, an accurate estimation of the gas concentration is also of interest. In simulation the ground truth gas concentration is available and allows us to quantify the error in the estimated gas concentration. Therefore, in this section we are using simulations to evaluate and parametrize the proposed DARES strategy.

Our goal is to get an empirical understanding of the different parameters and to make sure that the proposed DARES strategy is able to find the gas sources. The finding in these simulations will allow us to parametrize our algorithm in real-world scenarios later on. In the next section we first present the used simulation setup in more detail. After that we will present and discuss the results of the parameter evaluation. The results and discussion in this chapter were originally published in [188].

7.1.1 Simulation Setup

For simulating the gas dispersion we are solving the forward problem of the advection-diffusion PDE (3.1). This means we treat the source distribution and wind velocity as known and calculate the resulting gas concentration. In the simulation we have full control over the wind condition and the number and location of sources. The simulated gas concentration is used to simulate pointwise synthetic measurements of the robot at requested locations in the environment. In most of the cases in this section we consider five robots (i.e. $L = 5$)

guided by the proposed our DARES strategy of Section 6.2. Their dynamics and other constraints are neglected and we assume that the robots can reach every desired point in our environment within infinitesimal small time and without any collision. Further, we consider the time to carry out a measurement to be zero. These simplifications allow us to study purely the performance of DARES that is not disturbed by side effects and is independent of the robotic platform or sensor.

The simulated measurements are plugged into the probabilistic model described by equation (4.1) in order to estimate the gas concentration and source distribution as well as the uncertainty map. From the implementation perspective, it means that in every update loop of the model according to Figure 6.3, five new measurements are incorporated.

The simulated gas concentration provides us with ground truth values for calculating the error of the estimated gas distribution. Similarly, the source distribution used to simulate the gas dispersion is used to quantify the error of the estimated source distribution. During this chapter we are going to evaluate the performance of the DARES strategy by means of two different error metrics which compare the estimates with the ground truth values. For the estimated gas concentration we look at the NMSE. Since we expect the estimated source distribution to be sparse, we measure its difference to the ground truth with the EMD [151]. Both metrics are explained in the Appendix C.

In practice, the FEM approach of Section 3.3.2 is used to numerically simulate the concentration. Based on equation (3.38) we can calculate the gas concentration $\vec{f}[n]$ with predefined wind velocities $\vec{v}_1[n]$ and $\vec{v}_2[n]$, and source distribution $\vec{u}[n]$. The matrices of equation (3.38) are assembled based on the FEM software FENICS [109]. Note that since the wind velocity is treated as known, equation (3.38) is linear. To have a well posed problem, additional boundary conditions are needed. In our case we selected a Dirichlet boundary condition which corresponds to an open field scenario. As an initial condition, we set $\vec{f}[0] = 0$ (all source start at time $n = 0$). For our parameter evaluation we assume a rectangular environment and choose to use a discretization mesh as a grid with 26x26 nodes. This results in 676 nodes. For gas simulation we consider a unit-less concentration and source strength. The diffusion coefficient κ in Equation (3.38) is set to 1 and the distance between two nodes in the mesh is also unit-less set to 1, without loss of generality. By using these normalized values we minimize possible numerical issues. The time discretization Δt is part of the simulation studies in the following. For some simulations we consider only the steady state of the PDE, i.e. equation (3.39).

The robots are guided by the DARES strategy described in Section 6.2. In short, locations with a high uncertainty in the gas concentration are preferred as new measurement locations. In order to calculate the uncertainty, the implementation described in Section 5.2 is used. The numerical optimization in equation (5.41) and (5.42) requires searching for 2304 unknowns in total at each time iteration. This is so since we are looking for the concentration, source

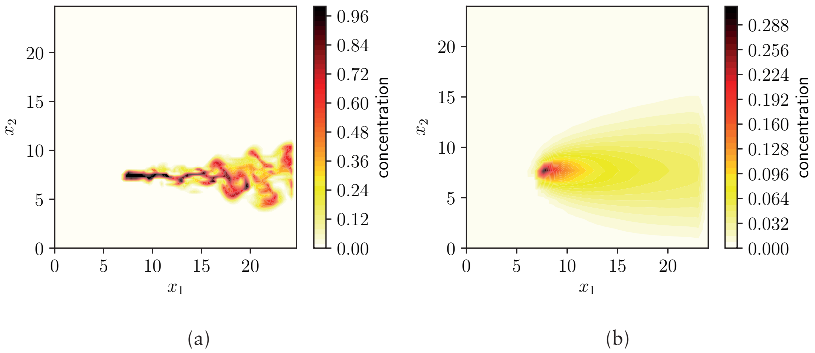


Figure 7.1: The figure compares the simulated gas distribution based on mantaflow in (a) and the advection-diffusion PDE in (b). While in (b) the space discretization is 26x26, in (a) the resolution of the simulation is 96x96 for a nicer visualization (linear interpolation between nodes in both cases).

strength and two wind components at each node. We would like to remark that we treat the boundary condition as known in the model for the exploration. Thus, the nodes at the border are accounted as known values.

With the simulation setup described so far, we are able to analyze the effect of model parameters such as the relaxation τ_s, the influence of our sparse prior or wrong wind measurements. However, for simulating the environment we use exactly the same model as for the DARES approach. For evaluating the robustness of our approach regarding model mismatch, we need a different simulation that is not based on the PDE (3.1). We therefore used the fluid/smoke simulator mantaflow [172], which accounts for turbulence effects. The software is based on solving the Navier-Stokes equations and is able to simulate small scale turbulence and vortexes, which we are specially interested in. As an example, Figure 7.1 shows the instantaneous simulated gas distribution in (a) compared to a simulated distribution based on the advection-diffusion PDE in (b). However, mantaflow primarily targets realistic smoke dispersion in the context of computer animations. It does not solve the full Navier-Stokes equations in DNSs. Instead, the RANS equations are solved and wavelets are used to model small scale turbulence. Therefore, the physical accuracy of the simulated gas dispersion process is limited. This does not matter so much for our purposes since we are interested in the effect of model mismatches and therefore our intention is to use a simulation that differs from our model assumptions.

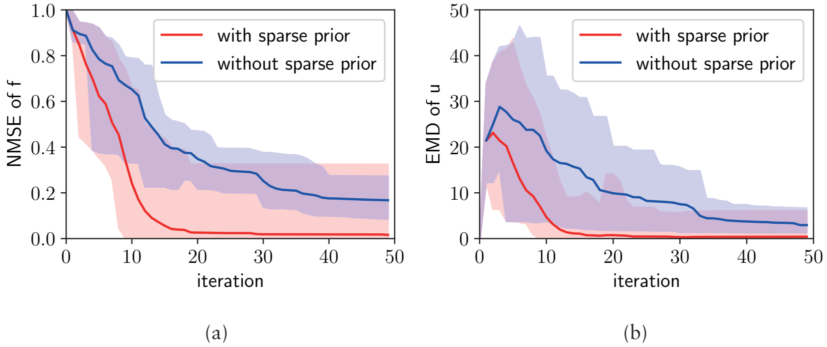


Figure 7.2: Comparison of the exploration performance: The plots in (a) and (b) demonstrate the impact of the sparsity assumption on the source distribution. In (a) the error of the estimated concentration field is shown as NMSE. In (b) the error of the source distribution is plotted by means of the EMD. The curves are averaged over 45 simulation runs, where as the shaded areas are limited by the best and worst case. (Simulation setup: advection-diffusion PDE; steady state; uniform, constant wind)

7.1.2 Parameter Evaluation

In the following we will study the influence of different parameters and adjustments of the proposed DARES strategy and their impact on the performance. These studies were partially published in [188].

Effect of Sparsity Inducing Prior

One of the main aspects proposed in this thesis is the sparsity inducing prior. It enables us to model the a-priori assumption that we do not know the exact number of sources, but we know that they are sparsely distributed. Now, we would like to analyze what we gain from introducing the sparsity assumption and how it affects the source localization performance. To this end, we carried out a set of experiments where we used a constant and uniform wind distribution with $\vec{v_1} = 1$ and $\vec{v_2} = 0$. Further, only the steady state (3.39) is considered. This makes the analysis easier, since no time dependencies have to be considered. For analysis, we averaged the performance over several simulation runs. In the individual runs we varied the number and the position of the sources, the relaxation parameter τ_s and the measurement noise τ_m (see Chapter 4 for the definition of these parameters). More precisely, we carried out 45 simulations covering all possible parameter summarized in Table 7.1.

source(s) at	$(10,10)$	$(15,13)$	$(10,17)$ $(17,7)$	$(8,12)$ $(15,12)$	$(12,17)$ $(5,10)$ $(15,12)$
τ_s	10^4		10^5		10^6
τ_m	10^4		10^5		10^6

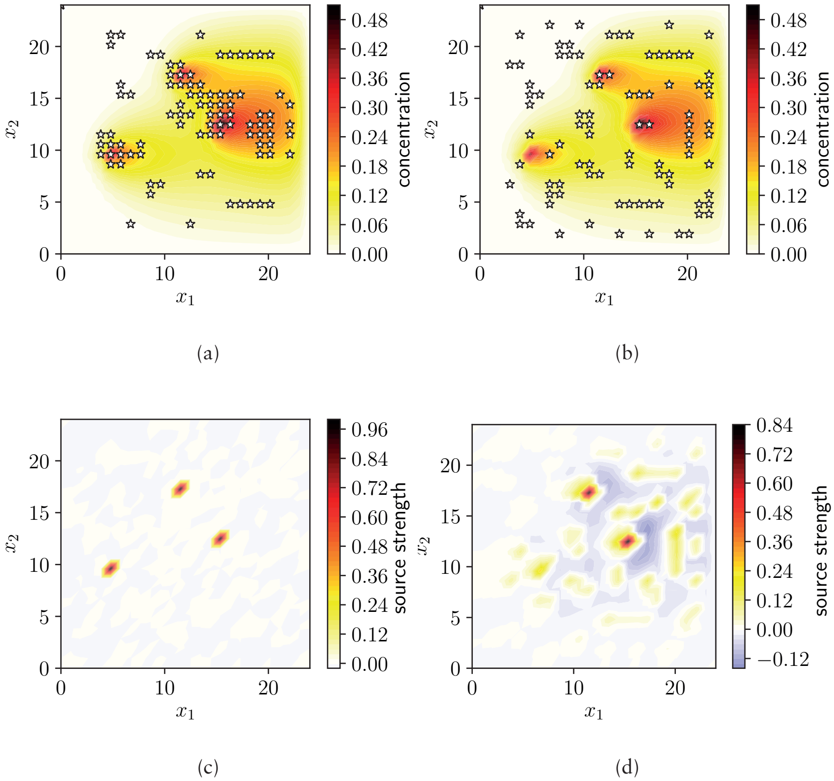
Table 7.1: Parameters used for simulation runs shown in Figure 7.2. The simulations cover all 45 (5x3x3) combinations of values for the three parameters.

The results in Figure 7.2 show the NMSE of the estimated concentration in Figure 7.2a and the EMD of the estimated source distribution in Figure 7.2b. The 45 simulation runs of the exploration were performed two times: once with sparsity inducing prior and once without the prior assumption. From the plot we can clearly see, that the performance of the exploration is better in case the sparse prior assumption is used: the error is reduced faster, i.e. lower error values are achieved with fewer iterations. Note that after each iteration five new measurements were added to the estimation problem. From the EMD plot in Figure 7.2b we can conclude that on average after 20 iterations the sources are correctly identified. From a numerical point of view, this means we were able to estimate the 576 elements of $\vec{u}$ with only 100 measurements.

As a possible intuitive explanation of this observation, let us have a look at the example in Figure 7.3 where the sampling locations are marked by white stars. When inspecting the sampling pattern of the DARES strategy more closely, we can see that the proposed DARES strategy prefers regions close to potential sources in case when the sparsity prior is used. This pattern is the reason for the successful and faster identification of the exact source distribution. DARES prefers these locations close to sources, because every newly estimated source contradicts the assumption that the sources are sparsely distributed. As a consequence, the uncertainty at such a location rises and attracts more measurements. In contrast, without a sparsity inducing prior, the samples end up being approximately uniformly distributed over the environment as can be seen in Figure 7.3b.

Trust in PDE Model and Sensor Noise

The second important contribution of our approach is the relaxation of the model assumptions. With the precision parameter τ_s we parametrize our trust in the PDE model. From the results of the simulations we will see how this parameter influences the performance of the source localization. Actually, it is the ratio of the precisions τ_s to the precisions τ_m of measurement noise that plays an important role. For these simulations we have chosen the parameters according to Table 7.2. The simulations cover all possible 125 combinations



(a)

(b)

(c)

(d)

Figure 7.3: The two plots show as an example the sampling pattern of the DARES strategy with a sparsity prior in (a) and without the use of a sparsity prior in (b). The parameters chosen are $\tau_m = 10^5$, $\tau_s = 10^4$. The white stars indicate the measurement locations. The color map represents the ground truth gas concentration field in (a) and (b). The sources are actually located at the locations with the highest concentration in (a) and (b). The estimated source distribution is shown in (c) with the sparsity prior and in (d) without the sparsity prior. In both cases the scene is a snapshot after 18 iterations. (Simulation setup: advection-diffusion PDE; steady state; uniform, constant wind)

of parameter from this table. Again, we used a constant and uniform wind distribution with $\vec{v_1} = 1$ and $\vec{v_2} = 0$ and consider the steady state (3.39).

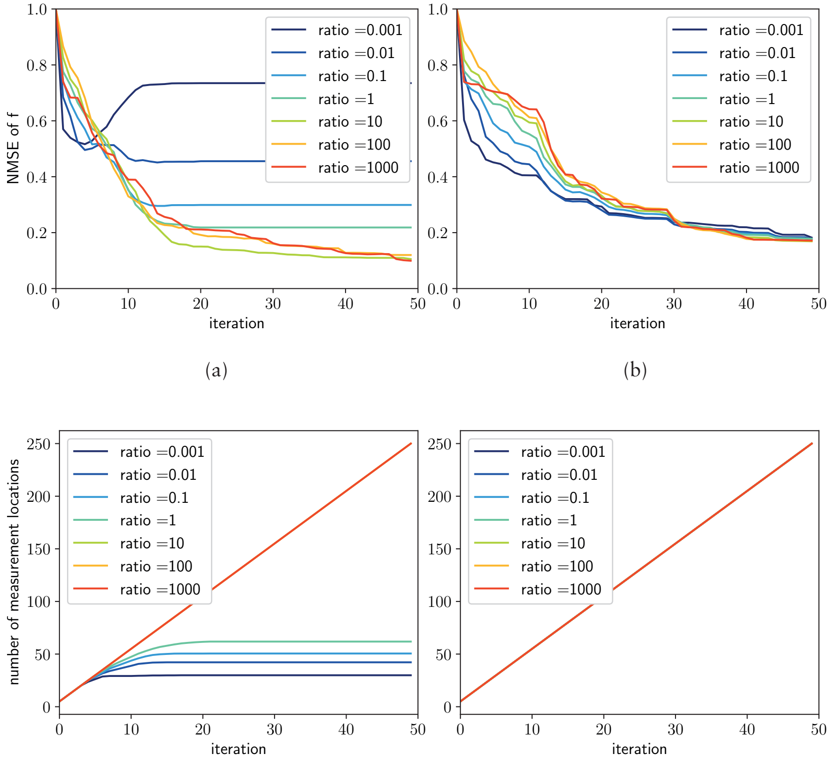
In Figure 7.4 compares different ratios of τ_m/τ_s. We averaged over all simulation runs where the respective parameter combination corresponded to a certain value of the ratio τ_m/τ_s. The plots in Figure 7.4a and 7.4b show the

source(s) at	$(10,10)$	$(15,13)$	$(10,17)$ $(17,7)$	$(8,12)$ $(15,12)$	$(12,17)$ $(5,10)$ $(15,12)$
τ_s	10^2	10^3	10^4	10^5	10^6
τ_m	10^2	10^3	10^4	10^5	10^6

Table 7.2: Parameters used for simulation runs shown in Figure 7.4

performance with respect to the NMSE. Once again, we compare the performance with (in Figure 7.4a) and without (in Figure 7.4b) the sparsity inducing prior. As can be seen, if no sparsity assumption is used, the performance is not as sensitive to the choice of τ_m and τ_s compared to the case with a sparsity inducing prior. Moreover, from Figure 7.4a it gets clear that a wrong choice may have a crucial impact on the exploration performance. In order to find out the reason for this, let us look more closely at the number of sampling locations. The number of measurement locations is shown in Figure 7.4c and 7.4d. Please note this is not the number of measurements. The number of measurements is increasing in every iteration by five, i.e. the number of sampling points per iteration. However, some sampling points may be visited more than once. Therefore, the number of measurement locations not necessarily increases by five per iteration. As can be seen from Figure 7.4d, without a sparsity inducing prior the number of measurement location increases linearly by five for each iteration. This means that each new measurement is carried out at a new location that has not been visited before. If the sparsity prior is used, this is not the case for samll ratios of τ_m/τ_s. Figure 7.4c shows that the number of measurement locations converges for some ratios.

A possible intuitive explanation of the convergence of measurement locations is as follows. The presence of an estimated potential source implies a tendency to a violation of the prior assumption, which assumes that only few sources are distributed in the environment. Evidence for a new source slightly contradicts this assumption. As a consequence the uncertainty in the respective region grows making it more favorable for further measurements. In general the new measurements in the region automatically reduce the uncertainty again and thus compensate this effect. However, in case of low ratios of τ_m/τ_s, the trust in the measurements is comparably low. In other words, sensor noise is assumed to be high and measurements are not considered highly informative. Therefore, evidence from measurements is not able to compensate the effect of the sparsity inducing prior. As a consequence, the DARES strategy will tend to allocate measurements always within the same region, and the number of measurement locations does not increase with time. The measurements are repeated only in these regions and it is difficult to find other sources and thus the true source distribution.



(a) (b)

(c) (d)

Figure 7.4: Effect of different ratios of τ_m/τ_s: On the left in (a) and (c) the case with a sparsity inducing prior is shown, the panels on the right (b) and (d) result without a sparsity inducing prior. The first row (a) and (b) depicts the exploration performance by means of the NMSE of the estimated concentration field. The second row (c) and (d) plots the number of measurement locations. Note that in each iteration five measurements are carried out, however eventually at the same locations as before. (Simulation setup: advection-diffusion PDE; steady state; uniform, constant wind)

From the different combinations in Table 7.2, we found that $\tau_m/\tau_s = 10$ seems to be a good choice. Further, we found the best exploration performance for the combination $\tau_m = 10^5$ and $\tau_s = 10^4$.

Time Discretization

So far we have considered only the steady state performance of our dynamic process. Now let us analyze the effect of the chosen time discretization in the numerical approximation of the PDE. Essentially, we examine the performance of the exploration for different values of Δt in equation (3.38), still performing five measurements in each iteration. So actually, Δt describes the inverse sampling frequency of our measurements. It is the effect of this sampling frequency that we would like to analyze here.

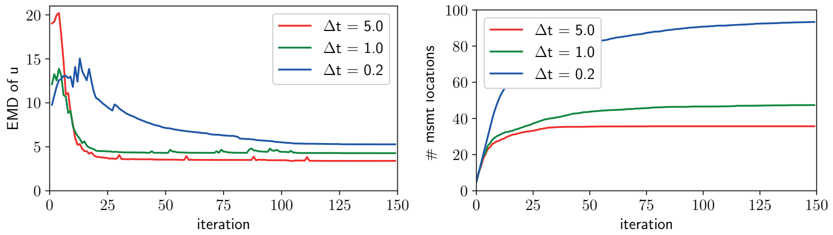
For this purpose we ran multiple simulations based on the advection-diffusion PDE for the dynamic case parametrized with all 90 combinations shown in Table 7.3. Apart from that we have chosen a constant wind field. For the dynamic case, an initial condition for the PDE is required. We have chosen the concentration field to be zero ($\vec{f}[0] = \vec{0}$) at the start of a simulation run. Figure 7.5 shows the corresponding results averaged over different combinations of parameters for fixed value of Δt. Analyzing and interpreting these results is quite difficult because different effects play a role and they cannot be studied separately.

source(s) at	$(10, 10)$	$(15, 13)$	$(10, 17)$ $(17, 7)$	$(8, 12)$ $(15, 12)$	$(12, 17)$ $(5, 10)$ $(15, 12)$
τ_s	10^3		10^4		$10^e 5$
τ_m		10^5		10^6	
Δt	0.2		1.0		5.0

Table 7.3: Parameters used for simulation runs shown in Figure 7.5

First, let us consider the effect of increasing Δt. Higher values correspond to more time in between measurements. So the dynamic process is faster or more dynamic compared to the sampling frequency. In other words, the gas spreads out faster with respect to the number of measurement iterations as in case of Figure 7.5. In contrast, a low Δt means that a lot of measurement iterations pass until the gas is spread out. Thus, in case of a low Δt, gas is mostly concentrated around the source and not widely spread during the first iterations. This makes it more difficult to find the sources. This effect may explain the fact that the exploration require fewer iterations for higher Δt; as can be seen in Figure 7.5a. However, if absolute time is considered, with a lower Δt (i.e. higher sampling rate) the error converges faster. Of course in case of a lower Δt, also more measurements are collected with respect to time.

Another effect that has to be considered in the dynamic case is aging of measurements. Because of the iterative update of the model in each time stamp n, the influence of a preceding measurement decreases over time. Note that previous measurements are indirectly encoded through the term $\vec{f}[n-1]$ in (3.38). If the time between two update iterations is higher (i.e. higher Δt), this aging



(a) (b)

Figure 7.5: Comparison of the performance with respect to different values of Δt, i.e. the time discretization of the PDE (note that Δt can be interpreted as the inverse sampling rate): The plots in (a) show the performance of the exploration measured by the error in the estimated source distribution by the EMD. In (b) the number of measurement locations is shown. Note that in each iteration five measurements are carried out, however measurements may be carried out at the same locations as before. (Simulation setup: advection-diffusion PDE; dynamic case; uniform, constant wind)

or forgetting of measurements has a stronger impact. As a consequence the DARES strategy tends to repeat measurements at locations, which have been measured already in the past. This effect can be seen in Figure 7.5b. Moreover, we would like to remark that also the identification of sources in combination with the sparsity inducing prior may influence the saturation of the number of measurement locations as we have seen in the previous section for different ratios of τ_m/τ_s. However, it is hard to separate those two effects.

Number of Robots

The proposed DARES approach is designed for a multi-robot system. Thus, the question rises, if we profit from using multiple robots compared to a single one. Therefore, in the following the source localization perfromance with respect to the number of robots is analyzed.

Until now, all performed simulations used five robots. Now we are going to evaluate different numbers of robots as summarized in Table 7.4. Let us recall that in the simulations the number of robots is reflected by the number of new measurements available in each update loop of the model according to Figure 6.3. We will investigate the effect of this number of new measurements per update loop. In the simulations we consider the static case (steady state) and the dynamic case of the diffusion equation (no wind). Up to four sources are placed at different locations according to Table 7.4 and the performance is averaged over these 10 different scenarios (in total 100 simulations).

source(s) at	(5,17)	(10,10)	(15,13)	(6,15) (15,8)	(10,17) (16,7)	(8,12) (15,12)	(12,17) (5,10) (15,12)	(17,7) (10,10) (19,15)	(10,17) (17,7) (5,5) (19,15)	(9,15) (12,8) (6,4) (17,17)
robots	1		3			5	7	10		
setup	static					dynamic				

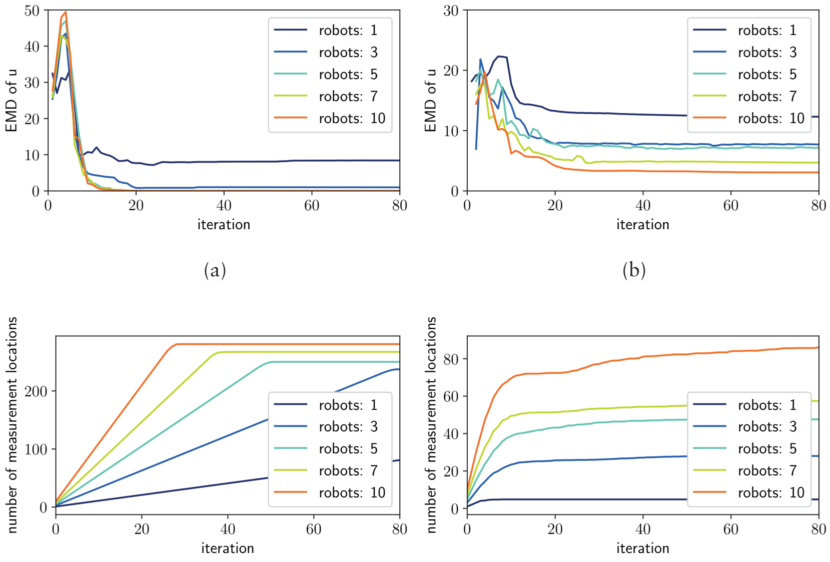
Table 7.4: Parameters used for simulation runs shown in Figure 7.6

The results are shown in Figure 7.6. Let us first have a look at the performance of the source estimate. For both cases - the static in (a) and the dynamic in (b) - the estimation error gets better with the number of employed robots. However, for the static case in (a) shows no benefit for employing more than five robots in our setup.

From the plots in (b) and (d) is can be seen that the number of measurements locations saturates after a certain number of iterations. Again, note that the number of locations is not to be confused with the total number of measurements. Of course the number of measurements increases in every iteration, but some locations maybe measured multiple times. Actually, as we have seen when studding the sparsity inducing prior the robots focus their measurements at locations close to an estimated source. This can cause issues if fewer robots than sources are used: A single robot will concentrate its measurements around the location of the first source that was found, which would prevent it to explore other regions and find other sources. This effect is especially strong in the dynamic case where measurements are aging (see also previous section). For further details on the effect of the number of robots we refer to [190].

Another interesting effect can be observed in the static case in Figure 7.6a. While the final estimate of the sources is worse for fewer robots, the convergence rate is quite similar for all robots: the estimates converged after approximately 20 iterations. However, it has to be noted that 10 robots actually carried out 200 measurements at this stage, while five robot only used 100 measurements and a single robot only 20. This effect can be ascribed to a disadvantage of the proposed DARES strategy. In the simulation the new measurements are carried out simultaneously at locations with the highest uncertainty at the moment. For example in the case of 10 robots, 10 locations with the highest uncertainty are chosen as the set of most interesting locations. However, if a measurement would be carried out at one of these 10 locations, the remaining 9 locations are not necessarily the locations with the highest uncertainty anymore.

A possible solution to this problem, would be to search only for a single location with the highest uncertainty first (as for a single robot). Then, the model can be update based on a virtual or predicted measurement at this location. After the update of the model, the next location with the now highest uncertainty can be added to the set of interesting locations. Of course such a procedure comes with a higher computational cost, since in every iteration the



(a)

(b)

(c)

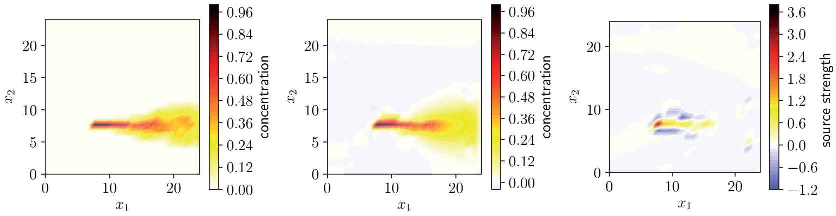
(d)

Figure 7.6: Effect of the number of robots on the exploration performance: In the first row (a) and (b) the EMD is shown and in the second row (c) and (d) the number of measurement locations for different numbers of robots is displayed. The left column (a) and (c) show the static case, where the right column (b) and (d) consider the dynamic case. (Simulation setup: diffusion PDE; dynamic and static case; no wind)

model has to be update several times in order to construct the set of interesting locations. In the real-world experiments this issue is less significant. There, measurements of the robots are carried out asynchronously and with a lower update rate compared to the update of the model. In this way, the model and the set of interesting locations is updated more often than measurements are taken. We will see this effect already in the Hardware-in-the-Loop experiments in Section 7.

Model Mismatch

This thesis proposes the DARES approach for gas source localization which relies on a model of the gas dispersion process. Therefore, we are also interested in the effect of model mismatch on the source localization performance. A model mismatch occurs when the used PDE model does not approximate the

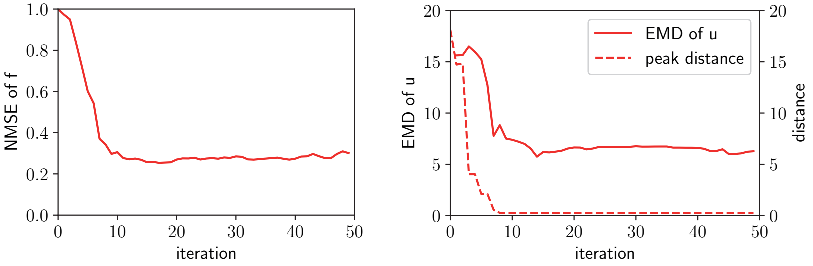


(a) (b) (c)

Figure 7.7: Exploration of a gas distribution simulated with mantaflow after 10 iterations (50 measurements): Plot (a) shows a snapshot of the gas concentration field simulated by mantaflow and averaged over 20 seconds. In (b) the estimated concentration field is shown and (c) depicts the estimated source distribution.

true physical gas dispersion process accurately. Instead of using the same PDE for forward-simulation and DARES, we thus simulate the ground truth gas dispersion using the mantaflow smoke simulator. The mantaflow smoke simulator models realistic flow dynamics in more details and allows us to generate a situation that we expect also in real-world applications. In contrast, the PDE model that is used by the DARES strategy makes use of a stronger approximation of the gas dispersion. It models important components of fluid flow, but not all its complexity. Using a simplified model allows us to exploit domain knowledge and to constrain the computational complexity but, of course, it is important to know the consequences of this model mismatch. To address this question, we simulated a single source. An exemplary snapshot of the instantaneous concentration field is shown in Figure 7.1a. Measurements taken during exploration are averaged over 20 sec. In this way we take the typically slow sensor response of gas sensors into account. Please note that, as a side effect, this smoothes out rapid fluctuations caused by turbulence in the gas dispersion. The PDE process model used in the exploration is parametrized as follows. We use five robots and the wind prior is set to 2 for the component in x_1-direction (0 for x_2-direction). The precision τ_s is selected from $\{10^3, 10^4, 10^5\}$ while the ratio $\tau_m/\tau_s = 10$ is fixed. Moreover, we consider a steady state of the gas dispersion process.

The results are shown in Figure 7.7 and 7.8. As expected, the performance of the estimate is worse compared to the results without model mismatch. Both, the NMSE of the estimated concentration and the EMD of the estimated source distribution is relatively high. To illustrate the quality of these values, Figure 7.7 shows a snapshot of the estimation during the exploration after 10 iterations ($\tau_s = 10^4$ and $\tau_m = 10^5$), i.e. 50 measurements. Figure 7.7a shows the con-



(a)

(b)

Figure 7.8: Performance of our DARES strategy when a different model (mantaflow) is used for simulation and for exploration: the plot in (a) shows the NMSE of the estimated concentration field. In (b) the EMD of the estimated source distribution is plotted. Besides, the Euclidean distance of the location of the simulated source and the peak in the estimated source distribution is shown.

centration field averaged over 20 sec. These values from this field are used as measurements in the simulation. Figure 7.7b depicts the estimated concentration field and Figure 7.7c the estimated source distribution, respectively. While we would expect the source distribution to be a single peak, it can be seen that the actual estimated source distribution is a wider shaped distribution. This is the reason for the rather poor EMD value. However, in this simple case with a single source we can also calculate the distance of the maximum of the source distribution (peak) as the most likely location of the source from the ground truth location of the source. This is plotted in Figure 7.8b. There we see that despite the shape of the source distribution is not a simple peak, the location of the source is correct.

To sum up, we observed a certain drop in the performance of the estimate of the source distribution based on the DARES strategy when we change to the mantaflow simulator. In general we would expect a similar behavior in a real-world application. Whether the estimated location of the source is good enough to claim that the source was successfully found, depends on the requirements of the particular application.

7.2 Experimental Infrastructure

In the previous section we evaluated the DARES strategy purely in simulations. In the rest of this chapter we are going to evaluate the DARES strategy in experiments. The experiments require appropriate gas sensors, robotic platforms and other infrastructure, such as a precise localization system as well as a communications system. All these components will be discussed in this section.

7.2.1 Sensors

The experiments use ethanol as a “toy” gas, because it does not imply any threats for the scientists conducting the experiments nor does it cause environmental pollution. In order to measure the ethanol vapor, two types of sensors were used.

Metal Oxide (MOX) Gas Sensor

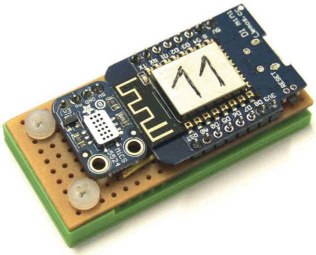


Figure 7.9: MiCS5524 MOX sensor with microcontroller.

The first sensor is the low-cost MiCS5524 MOX sensor from SGX Sensortech Ltd that is sensitive to VOC, carbon monoxide and ethanol [1]. MOX sensors consist of a heating element and a grain, semiconducting surface. In presents of a reducing gas, the resistance of the surface layer decreases, since the potential barrier is lowered [65, 102](see also Section 2.1.2). In general MOX sensors suffer from long response times and even longer recovery times, which cause a limited time resolution [155]. Further, they show long and short term drift [96], which requires frequent recalibration. However, they are inexpensive and highly sensitive. Let us also remark that in the planned experiments we are looking only for one known substance. Thus, the capability to select between different gases is not important in our setting.

The change in the resistor of the MOX sensor proportional to the gas concentration is measured by an Analog-to-Digital Converter (ADC) of a microcontroller. The data sheet of the sensor [1] specifies a linear dependency in the log-log domain between the sensor’s resistor R_s and the ethanol concentration. Therefore, we can model this dependency by an exponential function [81] as follows:

$$y_l = \beta_l \cdot R_s^{\alpha}, l = 1, ..., L \tag{7.1}$$

where y_l is the measured gas concentration plugged into our Bayesian approach (Note: L is the number of robots, i.e. number of gas sensors). The parameter α in (7.1) corresponds to the slope of the sensor sensitivity in the log-log domain. According to the data sheet [1] $\alpha = -1.6$. The parameter β_l was chosen in such a way that y_l is normalized to be unit-less and roughly in the range of $[0, 1]$). The parameter β_l is individually adjusted in a calibration step for the sensor of each robot $l = 1, 2..., L$. In this way the sensor responses are roughly the same across the swarm when exposed to the same constant concentration. The resistor R_s was calculated according to the measured voltage drop U on a load resistance R_L of a voltage divider consisting of R_s and R_L as follows:

$$R_s = R_L\,(5V/U - 1) \quad \text{and thus} \quad y = \hat{\beta_l}\,(5V/U - 1)^{\alpha}\,. \tag{7.2}$$

It is this voltage U that is actually measured by the microcontroller. We would like to remark that we incorporate R_L in the constant $\hat{\beta_l}$ when inserting (7.2) into (7.1).

Photoionization detector (PID) Sensor

The second sensor employed in the experimental evaluation is the Photoionization detector (PID) Sensor PID-AH2 from Alphasense [10]. It is sensitive to a lot of substances, among others also to ethanol. The PID Sensor consists of a high energy ultraviolet lamp. The lamp transmits photons through a chamber containing the gas of interest. The photons are absorbed by the molecules in the chamber and this photoionization generates electrically charged ions. The ions are attracted by an electrical field generated between two electrodes, which causes an electric current proportional to the amount of molecules, i.e. the gas concentration [14] (see also Section 2.1.3). In case of the PID-AH2 Sensor, the concentration is provided as an analog output voltage. It is important to note that the photon energy emitted by the lamp has to be higher than the ionization potential of the gas. In our experiments we are using a Krypton Lamp emitting photos with 10.6eV [14]. This is sufficient to detect ethanol [16].

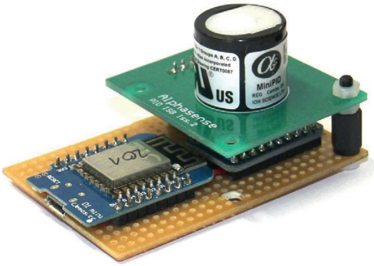


Figure 7.10: PID sensor with microcontroller

The main advantages of PID sensors compared to MOX sensors are their fast response characteristics and a linear relation of concentration and output voltage. Further, they are stable for a long time and do not suffer from drift.

In addition, the power consumption is quite low ($< 100mW$ for the PID-AH2) and they are light weighted ($< 8g$ for the PID-AH2) [10]. Thus, PIDs are very suitable for robotic gas source localization or gas mapping experiments.

In our experiments, the output voltage of the PID-AH2 sensor is sampled by a microcontroller with an ADC. The measured gas concentration y_l for the robots $l = 1, 2..., L$ plugged into our measurement model is linear proportional to the measured voltage as

$$y_l = \beta_l(U - \alpha_l) \tag{7.3}$$

The offset α_l caused by sensor characteristics and the ADC must be calibrated in clean air. The scaling factor β_l is likewise calibrated, such that y_l is unitless and roughly in the range of $[0, 1]$. Since the sensors show all a very similar characteristics the calibration factors turned out to be the same for the sensors of each robot.

Anemometer

The experimental evaluation in this thesis requires a sensor to measure the air flow (wind). Both, wind speed and wind direction, are needed in order to build an air flow model as described in Section 4.4. For indoor scenarios the air flow is expected to be quite low (e.g. $0.1 - 1m/s$). Based on these requirements an ultrasonic anemometer was chosen, namely the model 8600 from the R. M. Young Company [6]. It is a two dimensional anemometer which provides the wind direction only in the horizontal plane. Ultrasonic anemometers consist of several acoustic transmitters and receivers. By measuring the time of flight of sonic pulses they are able to infer the wind velocity. An ultrasonic anemometer has the advantage compared to mechanical devices that it measures low wind speeds more accurate and reacts faster to changes in the wind velocity.

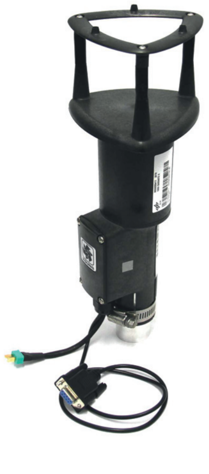
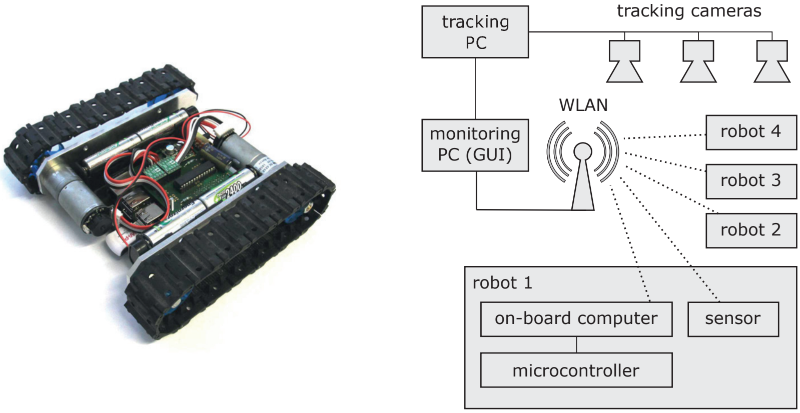


Figure 7.11: Anemometer

In our experiments the anemometer was either mounted on static tripods or mobile robots (e.g. see Figure 7.11). In case the anemometer is mounted on a robot, the robot's orientation has to be taken into account to translate the wind velocity into a global coordinate frame.

7.2.2 Robots and Navigation

The proposed DARES approach is evaluated in an indoor scenario in a laboratory and in outdoor scenarios. Depending on the location of the scenario different robotic platforms and infrastructure are employed. However, in all cases the software implementation of all required components is build up on Robot Operating System (ROS) [3].



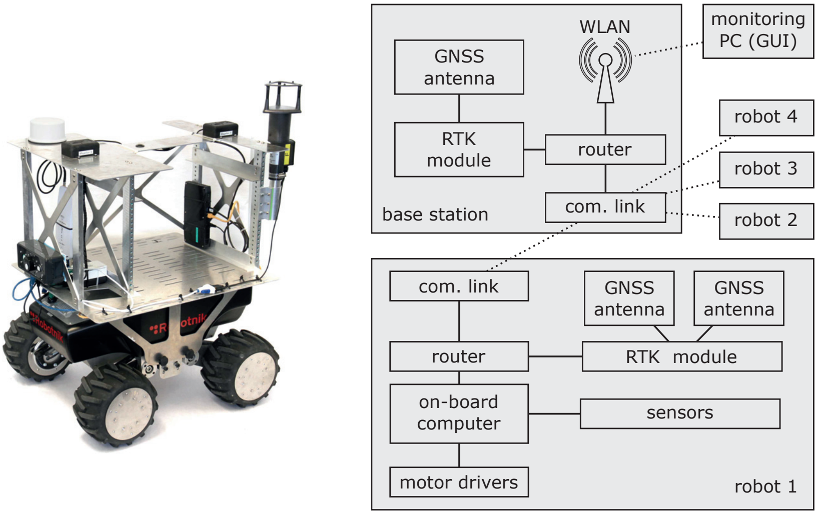
(a) (b)

Figure 7.12: The picture in (a) shows a small rover used in indoor experiments. The schematic in (b) depicts the system architecture of the infrastructure used in the indoor experiments.

Laboratory Scenario

For the laboratory scenario, custom-built ground-based robots are used as depicted in Figure 7.12a in order to carry the gas sensors. They are all equipped with low power on-board computers (Raspberry Pi 2900MHz quad-core ARM Cortex-A7 CPU, 1GB RAM) supplying Linux OS. We will show that even with these low power computers it is possible to perform all calculations required for our DARES approach. This is possible due to the use of the distributed algorithm (see Section 5.1) where the computational load is shared among all robots. From the on-board computer it is possible to send control commands to a micro-controller that implements a velocity controller for two motors driving the tracks of the robot. The velocity commands are generated by a path planner running on the on-board computer and driving the robot to desired way-points.

This path planner as well as the collision avoidance mechanism described in Section 6.3 requires an accurate position and orientation. To this end, the laboratory is equipped with a commercial optical tracking system. Using active infrared LEDs mounted on the robot the tracking system is able to compute accurately the robot's location and orientation in real-time. It can be considered as an accurate and reliable localization (error $< 1\text{cm}$) with respect to the discretization of the environment with a cell size larger than 10cm. Further, the robots are able to exchange data among each other through a WLAN communication link. Also, through this link the robots receive their current positions and orientation from the tracking system. The gas sensors mounted on the robots provide their measurement to the whole system through WLAN, too. A central PC enables a scientific operator to start and monitor the whole system. Moreover, the central PC shows intermediate results of the estimated source and gas concentration on a Graphical User Interface (GUI). The schematic in Figure 7.12b depicts the whole setup.



(a)

(b)

Figure 7.13: The picture in (a) shows the rover used in outdoor experiments. The schematic in (b) depicts the system architecture of the infrastructure used in the outdoor experiments.

Outdoor Scenario

For the outdoor scenario, a commercial robotic platform is modified for the planned experiments. In particular, it is the Summit-XL robot from Robotnik [148] modified as shown in Figure 7.13a. The robot is equipped with a powerful Intel NUC (i7-4790S CPU 3.20GHz, 8GB RAM) computer for onboard processing. Additionally, the robots are equipped with an anemometer and gas sensors, which are directly connected to this on-board computer. For navigation and path planning the open source navigation stack of ROS is used. The path planner makes use of the Time Elastic Band approach [150] to generate collision free trajectories through a static obstacle map (see also Section 6.3). The robots carry a Real Time Kinematic (RTK) Global Navigation Satellite System (GNSS) system which not only provides accurate position information (error $< 10cm$) but also the heading of the robot without the need of a magnetic compass. This GNSS position and orientation is further fused with the robot's odometry data. The on-board RTK requires real-time correction data from an RTK base station. Communication among the robots and the base station is implemented by a dedicated communication link in the 5GHz WiFi band. The whole system can be controlled and monitored with a central PC (Laptop). The schematic in Figure 7.13b depicts the whole setup.

7.3 Experiments

Based on the robots and infrastructure presented in the previous section, we carried out different experiments to evaluate the DARES strategy proposed in Section 6. In this section we first present results of Hardware-in-the-Loop experiments. Afterward, we evaluate the proposed air flow model and finally present results of experiments with ethanol vapor.

7.3.1 Hardware-in-the-Loop Experiments

Evaluation of gas mapping or gas source localization strategies in realistic real-world scenarios is challenging. The main reason is the lack of ground truth data to evaluate the performance of the developed approach. Of course ground truth data can be obtained for the source locations. However, the true spatial gas concentration distribution is generally unknown. In the previous section we circumvent the lack of ground truth data by using a simulator. While, in this way, it was possible to evaluate the performance of the developed source localization strategy, the simplified simulations do not account for real-world constraints like the robot's dynamics, communication limits, etc. To evaluate the impact of such constraints, this section evaluates the DARES strategy by means of Hardware-in-the-Loop experiments. Basically, we employ a real multi-robot system. However, the gas dispersion process is simulated as well as the synthetic gas concentration measurements taken by the robots. In this way we use ground truth data from simulation, while the DARES approach is still affected by real-world constraints, e.g. limited speed of the robots, delay in communications etc.

Indoor Small Scale Experiment

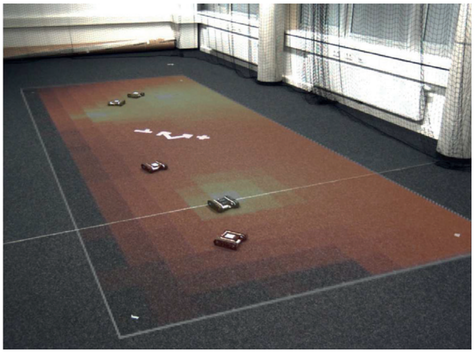


Figure 7.14: Laboratory environment: The picture shows five robots in the indoor scenario during an experimenter. The overlay depicts the simulated gas concentration.

The first setup for the Hardware-in-the-Loop Experiments is a small scale environment in a laboratory [1]. The five small robots described in Section 7.2.2 carried out the experiments. We make use of the DARES strategy of Section 6.2 and the robots were coordinated according to the collision avoidance mechanism described in Section 6.3.1 based on rescheduling. The size of the experimental environment was 2.4m x 6.0m. We used FDM and discretized the environment into 12x30 cells in order to simulate the gas dispersion process. In this experiment we consider a pure diffusion-based gas dispersion process, i.e. the wind field is assumed to be zero. Thus, the dynamic linear model of equation (3.40) is used to simulate the gas dispersion.

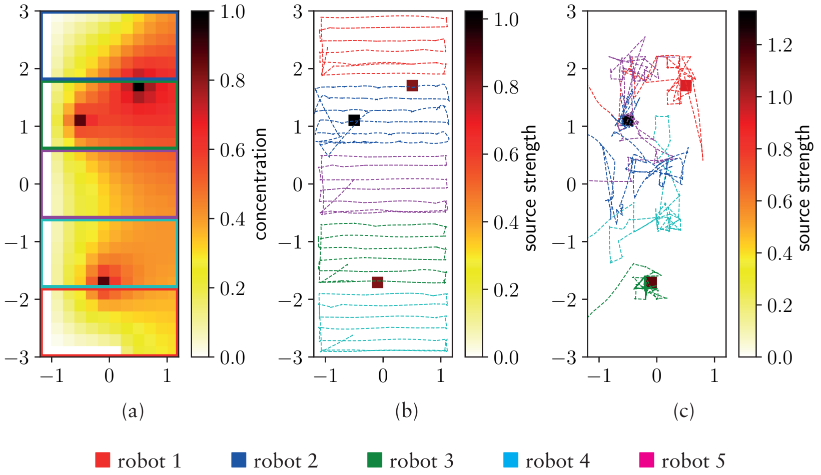
The simulated ground truth gas concentration is shown in Figure 7.14 and Figure 7.15a. The concentration is driven by three sources. One is placed at $x_1 = -0.2m, x_2 = -1.6m$ with source strength 1.0, the second at $x_1 = -0.6m, x_2 = 1.0m$ with strength 1.0 and the third at $x = 0.4m, y = 1.6m$ with strength 0.8. As can be seen, one source is isolated in the lower region of the exploration area, while the other two are more close to each other. The latter incurs some strong spatial correlation between them. Further, one of those is placed in an area with a generally higher concentration level.

We have chosen Dirichlet boundary conditions for all borders except for the right one, where we use a Neumann boundary condition. This corresponds to an open field scenario, where material can flow off at all boarders except for the right one (which represents e.g. a wall or a similar obstacle). Whenever a measurement is demanded by the DA̧RES procedure for a robot, equation (3.40) is evaluated at the current position of the robot in order to generate a virtual measurement. The concentration and source strengths are considered to be unit-less. The discrete grid size, the time difference between two discrete time stamps and the diffusion coefficient κ are all set to 1. However, in the experiments the concentration field is downscale to match our laboratory environment.

[1] Note: results were already partially published in [190] and [192].

The same model as in the simulator is used by the probabilistic formulation of the DARES strategy (see Section 4.2). While the simulator runs centrally, the DARES strategy is implemented in a distributed fashion according to Section 5.1. All calculations required to generate the way points for the robots are done on-board the robots cooperatively. To this end, the discretized environment is partitioned into 5 parts as shown in Figure 7.15a. Each robot is then responsible for the inference of parameters associated with one part only. Further, the robots also exchange messages across common borders.

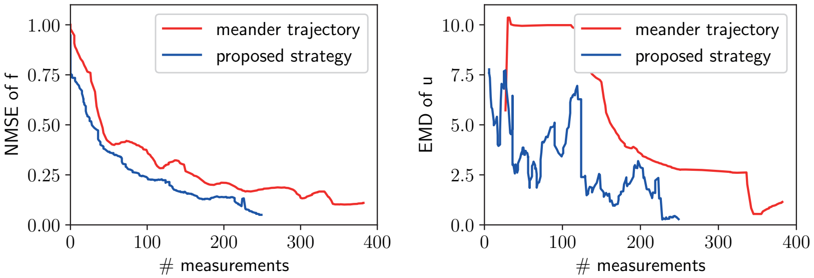
In this experiment, we compare the proposed DARES approach against exploration with a predefined sweeping trajectory for the case of 5 robots. The sweeping trajectories are generated by simply dividing the environment into five equal regions and generating a predefined “meander” trajectory for each of these regions. These trajectories will fully cover the whole environment after a certain time, i.e. each grid cell is measured at least once. We make use of the NMSE to quantify the error of the estimated gas concentration compared to the simulated ground truth. To compare the estimated source distribution we use the EMD criterion. Both metrics are explained in Appendix C.



(a) (b) (c)

(d)

Figure 7.15: Illustration of the Hardware-in-the-Loop experiments: In (a) the gas concentration is shown as well as the assignment of regions to robots indicated by the rectangles. The plots in (b) and (c) compare the meander trajectories in (b) with an example of trajectories resulting from DARES in (c). The trajectories are shown on top of the estimated source distributions (black and red pixels indicate the sources, i.e. peak in the source distribution).



(a) (b)

Figure 7.16: The two plots compare the performance of the meander trajectory and the DARES strategy. In (a) the error is measured regarding the estimated source distributions by means of the EMD. In (b) the NMSE of the estimated concentration field compared to ground truth is shown.

Figure 7.15b and Figure 7.15c present the trajectories of the meander and a typical example of the proposed DARES strategy. Let us remark that the very nature of DARES is adaptive, i.e., the algorithm will react to the measurements taken. As such, the trajectory is not deterministic and Figure 7.15c shows one experiment as an example. The generated trajectories are overlaid with the estimated source distribution after 240 measurements in Figure 7.15c and after 360 measurements in Figure 7.15b. In both cases the three sources were nearly perfectly estimated. Figure 7.16 depicts the performance of DARES and the meander. The curves in Figure 7.16a depict the NMSE between the estimated concentration field and the ground truth in relation to the number of collected measurements. Figure 7.16a shows the EMD error between the estimated source distribution and the true source distribution. As we see, using the meander trajectory the multi-robot system is able to identify the source distribution correctly after approximately 340 measurements. This is indicated by the drop of the EMD towards zero. As we already saw in Section 7.1, measurements close to the source are needed to successfully localize it. Unfortunately, this means that the performance of the meander trajectory highly depends on the position of the sources with respect to the meandering pattern. If they are already covered at the beginning of the trajectory, fewer measurements are needed. In contrast to the meander, the error of the DARES strategy converges after only 230 measurements as can be seen in Figure 7.16b. This indicates that robots were able to localize the sources with fewer measurements. As can be seen from the trajectory in Fig. 7.15c, this is since the measurements are concentrated around the source locations. Obviously, the corresponding measurements contain more information about the sources, which is the reason for the better performance of DARES. Based on Figure 7.16a, it can be seen that the estimated concentration field for both strategies reaches a low NMSE. Here the performance of the proposed strategy is better, too. However, subjectively speaking the difference is relatively small.

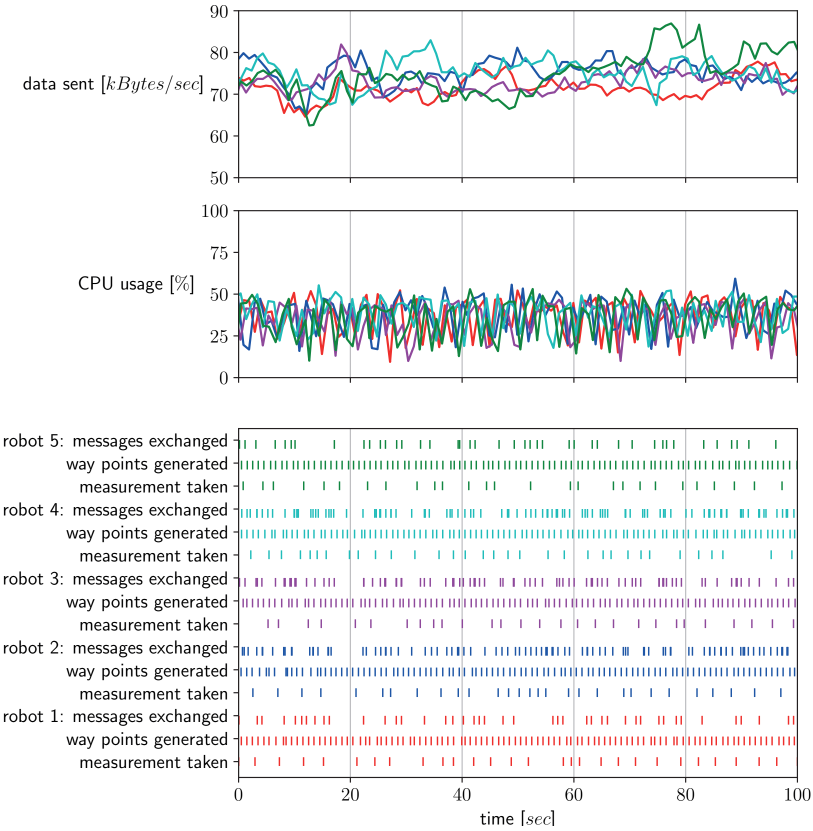


Figure 7.17: Depiction of the time line of a Hardware-in-the-Loop experiment as an example: The figure shows a time window of 100 seconds. At the top the data sent by the five robots are shown as well as the CPU usage of the robot's on-board computers. The lower part illustrates the time stamps when (i) messages are exchange with other robots, (ii) a new list of informative way points is generated by a robot and (iii) an actual measurement is taken by a robot.

The Hardware-in-the-Loop experiments further enable us to analyze other performance indicators of our distributed algorithm develop in Section 5.1.

Namely, we can measure communication data rates, the workload on the onboard computers and the frequency by which new interesting way points are generated. All these values are shown in Figure 7.17 for 100 seconds as an example. On average, less than $80kBytes/s$ are sent by the robots to their neighbors and the monitoring PC, including all overhead caused by ROS, TCP and the operating system. The low power on-board computers (see Section7.2.2) show less than 50% CPU usage. The lower plot in Figure 7.17 indicates the time stamps when (i) a robot exchanged messages with its neighbor according to the distributed message passing algorithm, (ii) a robot generated new way points as proposals for new measurement locations and (iii) a robot took a measurement. As can be seen, the robots are able to generate way point proposals approximately every second. This is sufficient, since the way points are more often generated than measurements are taken. The frequency of measurements depends mostly on the size of the environment and the robot's speed.

To sum up, the robots in the Hardware-in-the-Loop experiments are able to localize the simulated sources based on the proposed DARES strategy. In the experiments the robots are affected by numerous real-world constraints, which could not be considered in pure simulations described in Section 7.1. For example, the robots have to coordinate themselves and are affected by the collision avoidance mechanism which may prevent reaching the desired way points. The communication between robots is imperfect and affected by jitter, thus messages may be delayed. Nevertheless, the whole approach is robust enough to compensate for such effects and localize the sources correctly.

Outdoor Large Scale Experiment

As a second larger scale scenario, Hardware-in-the-Loop experiments were carried out in a gravel pit. The infrastructure and five robots as described in Section 7.2.2 were employed. The size of the environment was approximately $100m$ times $50m$. A static advection-diffusion gas dispersion process according to equation (3.39) is considered with $\kappa = 1$. This equation was used to simulate the gas concentration driven by two sources. The synthetic sources were placed at $x_1 = -10, x_2 = -10$ and at $x_1 = -50, x_2 = -20$ both with strength 1. Again, the source strength and the concentrations are considered to be unit less. Further, Dirichlet boundary conditions are used to model an open field scenario, where material can flow off at all boarders. The wind is considered to be uniformly distributed with $v_1 = -0.4m/s$ and $v_2 = -0.4m/s$ (see also Figure 7.19). The simulated gas concentration is used to generate synthetic measurements at the robots positions.

Our model-equation (3.39) was generated using FEM. The environment is discretized by a mesh consisting of 120 nodes (vertexes) and 183 elements (triangles). In Figure 7.18, the gravel pit and the five robots are depicted. The simulated gas concentration in the environment is shown as a colored overlay.

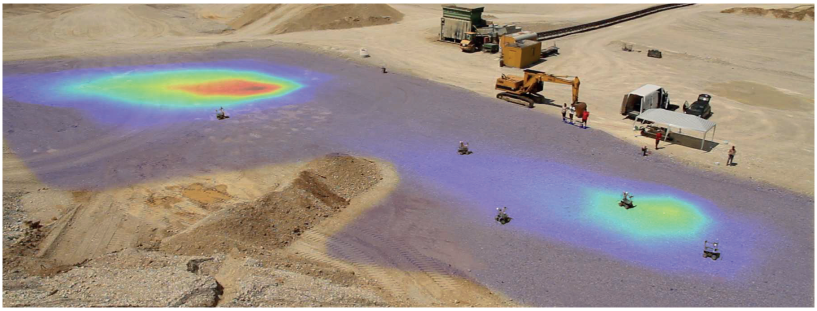
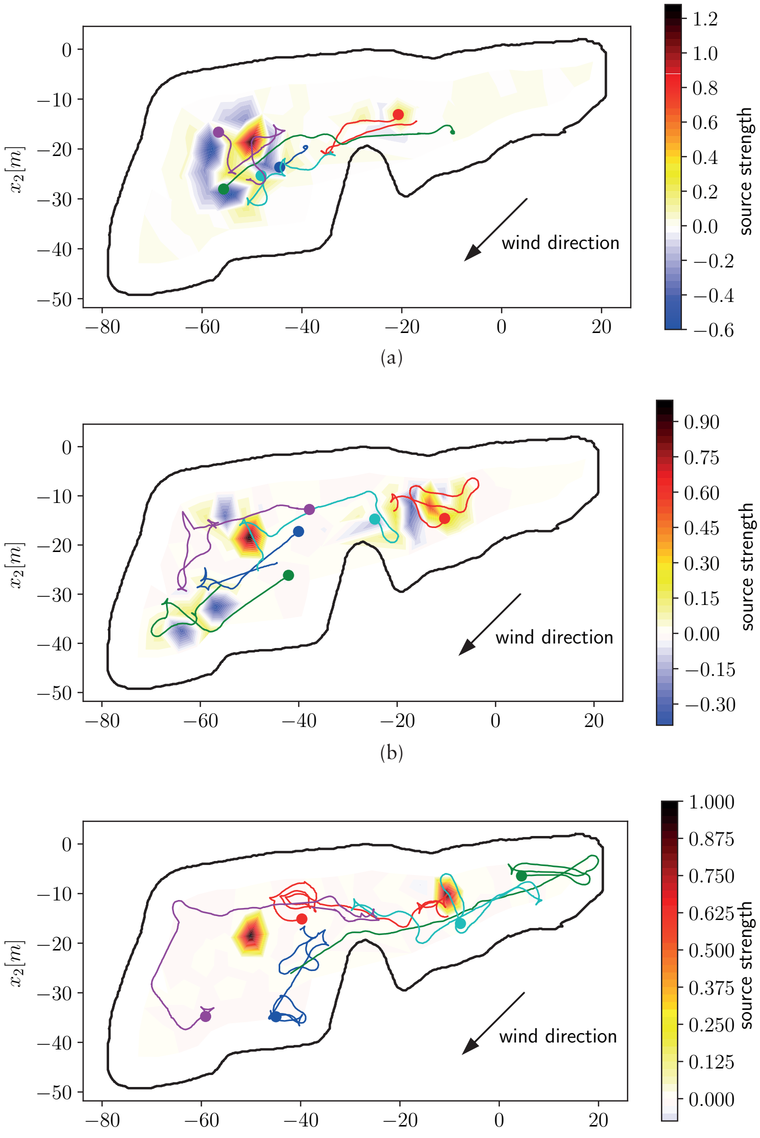


Figure 7.18: The picture shows the gravel pit where the outdoor experiments have been conducted. The color overlay represents the simulated gas concentration.

Since we consider a non-linear case here (advection-diffusion model), the DARES approach is implemented according to Section 5.2. Thus, the proposed interesting way-points are generated centrally on the monitoring PC (see Figure 7.13) and send to the robots. The robots are coordinated according to collision avoidance mechanism of Section 6.3.2.

The results for one example run are shown in Figure 7.19 and Figure 7.20. In Figure 7.19 three snapshots of the estimated source distribution are shown. These snapshots correspond to the three time stamps marked by dash lines in Figure 7.20, namely 65s for Figure 7.19a, 150s for Figure 7.19b and 300s for Figure 7.19c. As can be seen, the two sources are successfully identified after approximately 300 seconds, which is also indicated by the low EMD between the estimated source distribution and the ground truth source distribution. The EMD in Figure 7.20 shows the difference between the estimated source distribution and the ground truth source distribution which has been used to simulate the gas dispersion process (see also Appendix C). It can be seen that the EMD values are much higher as in the small scale experiments in Section 7.3.1. This effect is due to the very nature of the EMD, which depends on the scale of the spatial distributions that are compared to each other. Loosely speaking: in the small scale experiment, the mass of the source distribution has to be moved only a few meters to convert the estimated distribution into the ground truth distribution. In the large scale experiment this effort is higher, since the distribution mass has to be moved several meters.

The three time stamps in Figure 7.19 depict the trajectories of the five robots. The trajectories cover only the time window until the previous time stamp (i.e. in (a) $0-65$s, in (b) $65-150$s and in (c) $150-300$s). It can be seen that the robots first move to the lower left region and explore the upper right region at last. This behavior is reasonable considering that the wind direction is from upper right to lower left (indicated by the black arrow). The down-



(a)

(b)

(c)

Figure 7.19: The three plots show the estimated source distribution after 65s, 150s and 300s. In addition the robots and their trajectories in the respective time segment are plotted.

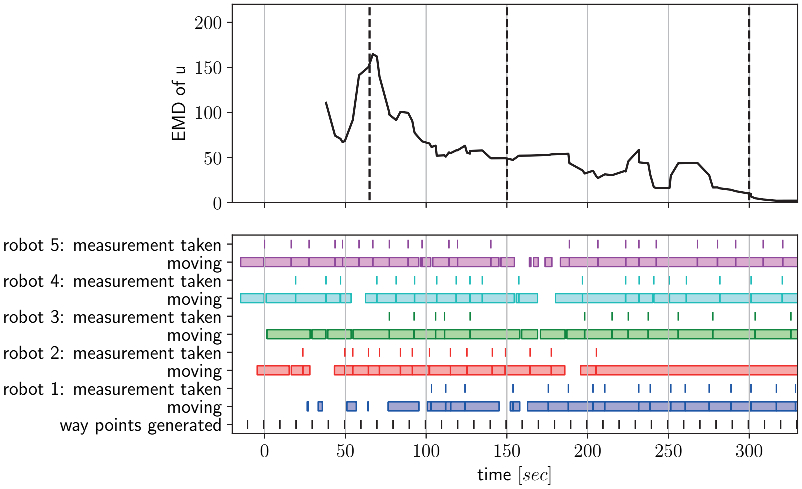


Figure 7.20: The figure illustrates the temporal course of the Hardware-in-the-Loop experiment. The upper part plots the EMD between the estimated source distribution and the ground truth source distribution. The lower part shows the time stamps when the robots took their measurements and the time windows during which the robots were moving.

wind region (lower left) is more informative compared to the up wind region. Consequently and reasonably, the robots start their exploration there.

Again this Hardware-in-the-Loop experiment allows us to investigate real-world constraints. Namely, we are interested how the collision avoidance mechanism and dynamic constraints of the robots affect the exploration. This can be studied by means of the time-line plot in Figure 7.20. The way points are generated by the DARES approach with a fixed update frequency of 10s. This update frequency is still high compared to the times when measurements are taken by the robots. The measurements are taken quite irregularly depending on the time the robot needs to move from one measurement location to the next. Further, the time line in Figure 7.20 indicates when robots are not moving. During these time windows the robots are blocked and are waiting for robots with higher priority to clear the way according to the collision avoidance mechanism described in Section 6.3.2. This collision avoidance mechanism has definitely a strong impact on the exploration performance. For example robot 1 is not able to take a measurement for the first 100s. Since the robot has the lowest priority, it has to stop for other robots and is therefore blocked for a long time. Even though the coordination may be suboptimal, the robots are able to find the sources in quite a short time ($5min$).

7.3.2 Air Flow Model

In the previous section we evaluated the proposed DARES strategy in Hardware-in-the-Loop experiments. Before, we move on to the real-world experiments in the next section, this section is going to evaluate the air flow model proposed in Section 4.4 experimentally. This evaluation includes the parametrization of the air flow model, which will be used in the real-world experiments. Again, two different scenarios are considered. The first scenario is a controlled indoor environment in a laboratory with an artificial air flow field. The second scenario is an open field outdoor environment on a windy day.

Indoor Scenario

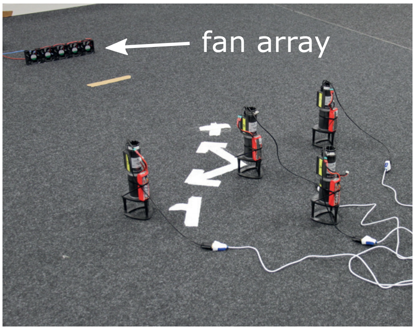


Figure 7.21: The picture shows five fans generating an artificial wind field. The field is sampled by four anemometers placed upside-down.

The indoor experiments are carried out in the same laboratory environment as the Hardware-in-the-Loop experiments in Section 7.3.1 (see also Figure 7.14). The considered environment is 3.0m times 5.0m large and discretized by 30 times 50 cells (edge length 0.1m) for the numerical approximation of the PDE, which makes use of the air flow model later on. The air flow is generated artificially by five small fans powered with 7.5V as shown in Figure 7.21. The fan array is located at $x_1 = 1.5\mathrm{m}$ and $x_2 = 0.0\mathrm{m}$. The air flow field was sampled by multiple anemometers (see Section 7.2.1) in order to obtain the required input data for the air flow model described in Section 4.4. We used 4 anemometers simultaneously to sample the air flow at uniform random locations in the environment. These sampled locations are marked by stars in Figure 7.22. As depicted in Figure 7.21, the anemometers are placed upside down to acquire data as close to the ground as possible. Figure 7.22a shows the air flow map resulting from one example data set. To be more precise: The air flow model predicts a mean of the x_1 and x_2 air flow velocity component at each cell in the environment according to equations 4.13. These two components are represented by the quiver plot in Figure 7.22a. Further, as an overlay, the predicted absolute air flow speed is shown. Please note that the model delivers not only the mean but also the variance for the two air flow velocity components, which are not shown here.

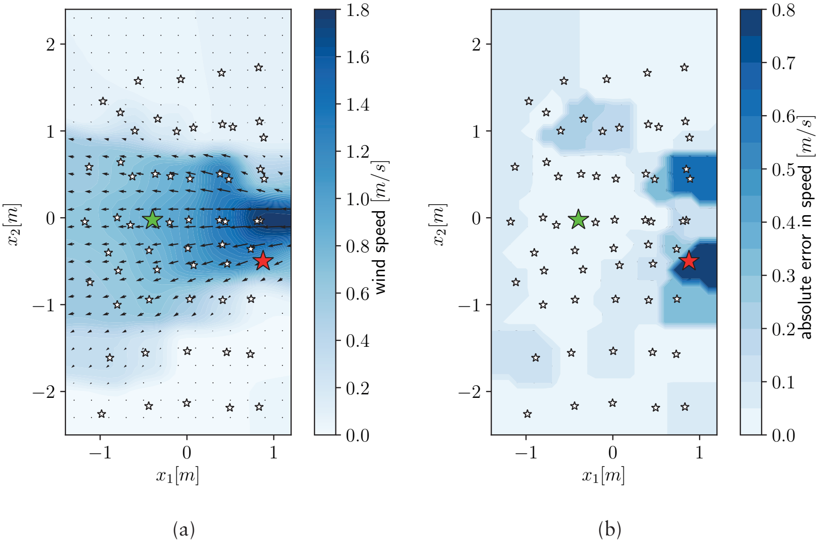


Figure 7.22: The figure shows in (a) the predicted air flow field. The white stars indicate the locations, where the air flow was sampled. In (b) the error of the prediction is plotted based on a leave-one-out cross validation. The green start indicates the location where the prediction is best and the red start were the prediction is worst.

Unfortunately, we lack of ground truth data to which we can compare our model predictions for evaluation purposes. Therefore, we make use of leave-one-out cross validation. As an example, let us consider the sampled location marked by the green star in Figure 7.22a. In Figure 7.23 the histograms of the measured air flow velocity components at the location indicated by the green star for x_1 and x_2 directions are plotted. Note that these components are actually calculated based on the wind speed and wind direction which are the outputs of the anemometers. To the measurements we fitted a Gaussian curve. In addition, Figure 7.23 depicts the predicted Gaussian PDF by the air flow model. For calculating this predicted Gaussian the measurements at the location marked by the green star were not considered. As can be seen, the two Gaussian curves coincide very well and the predicted PDF is similar to the data histogram.

Figure 7.22b maps the absolute error between the predicted mean and the actual mean of the air flow speed at the sampled locations. To generate this error map, for each sampled location a leave-one-out cross validation was applied. For a better visualization the errors are interpolated in between the sam-

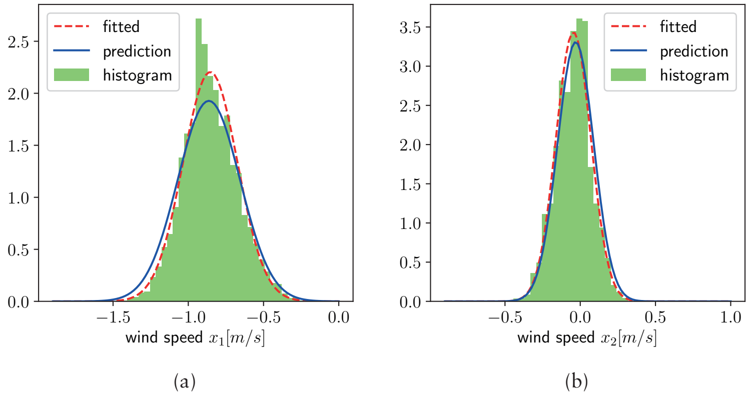
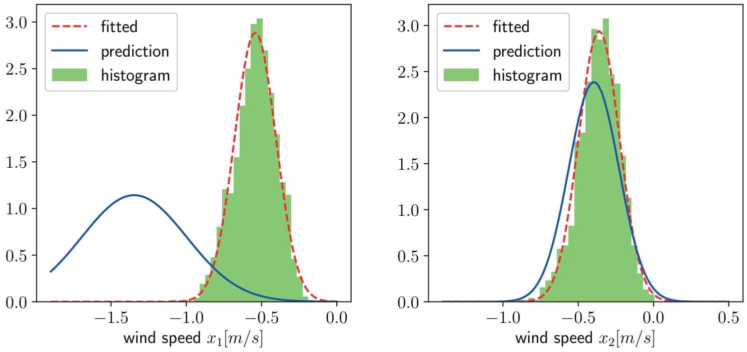


Figure 7.23: The figure compares the predicted air flow (blue curves) to the actual measurements (green histogram) at the location indicated by a green star in Figure 7.22. Further, a Gaussian curve is fitted to the histogram (dotted, red curve). In (a) the comparisons is shown for the x_1 component of the air speed and in (b) for the x_2 component.

pled locations (nearest neighbor interpolation). As can be seen, especially close to the fans the error is quite high and the prediction is not everywhere as good as at the green star location. If we look at the location marked by the red star in Figure 7.22a, the prediction is quite different from the actual measured data as indicated by the plots in Figure 7.24. Actually, this is the worst location in our example. In general, it can be observed that for regions with stronger gradients in the air flow speed the prediction is weaker.

Finally, let us remark on the used parametrization of the air flow model. Essentially, the model is fully defined by the two kernel widths k_s and k_t according to the model in Section 4.4. In our experiments we have seen that the air flow speed is very stable with respect to time, because of the controlled indoor environment. Therefore, we let k_t go to infinity. In this way, we neglect time dependencies of measurements in the air flow model and treat the air flow as a static process. In practice, $k_t \to \infty$ cancels out the second factor $e^{-\frac{2}{k_t}(n_i-n)^2}$ in weighting function (4.15). The choice of k_s is a little more involved. Again we compare the predicted Gaussian PDF and the Gaussian fitted to the measurement data by a leave-one-out cross validation as shown in Figure 7.23 and 7.24. For each sampled location the Kullback-Leibler divergence from the fitted Gaussian to the predicted Gaussian is calculated. In fact it is calculated two times: once for the x_1 air flow component and once for the x_2 component. All Kullback-Leibler divergences are summed up, i.e. for the two components at all locations. This sum indicates how well our model describes



(a)

(b)

Figure 7.24: The figure compares the predicted air flow to the actual measurements at the location indicated by a red star in Figure 7.22.

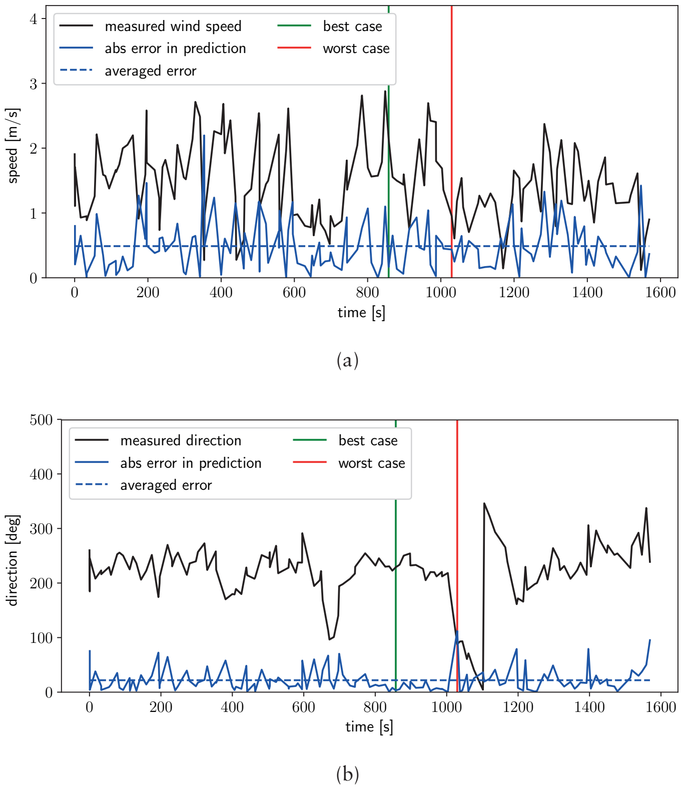
the data. The model parameter k_s is chosen so that the sum of Kullback-Leibler divergences is minimal. In our example, the optimal k_s is $0.11m^2$. In other words, the spatial kernel width of the weighting function (4.15) is approximately $0.33m$.

Outdoor Scenario

The experiments evaluating the air flow model outdoors took place on an open football field. The natural wind was measured by three robots equipped with anemometers (see also Section 7.2.1) and no artificial air flow was generated. The robots drove to random locations in the target area ($30m$ x $30m$) and waited at each location for 10 seconds. In the middle of this time window, wind measurements were collected for 8 seconds, while the robot was standing still. The robots were not synchronized, so most of the times while a robot was measuring the others were driving. In the following one experiment is shown and discussed as an example.

In Figure 7.25 the collected measurements are depicted for 25 minutes (black curves). More precisely, for every measured location the wind speed and wind direction averaged over the 8 seconds is shown. As can be seen by the strong variations in the wind speed, during the experiment the wind was not too strong, but “choppy” (i.e. recurrent increasing and decreasing wind speed). Therefore, predicting the air flow field is much more challenging than in the indoor environment.

In order to evaluate the air flow model we make use of a leave-one-out cross validation like for the indoor environment. For every sampled location we use the model to predict the air flow at this location and compare the prediction to the actual measurements. The prediction at a certain location does not incorporate the measurements taken at this particular location nor does it incorporate future measurements. Thus, the prediction can be calculated online and in real-time. Essentially, the air flow model predicts a Gaussian PDF — or rather its mean and variance — of the x_1 and x_2 air flow velocity component at each location in the environment according to equations (4.13) and (4.16). Calculated based on the mean x_1 and x_2 velocity component, Figure 7.25 plots the absolute error of the predicted wind speed and wind direction over time.

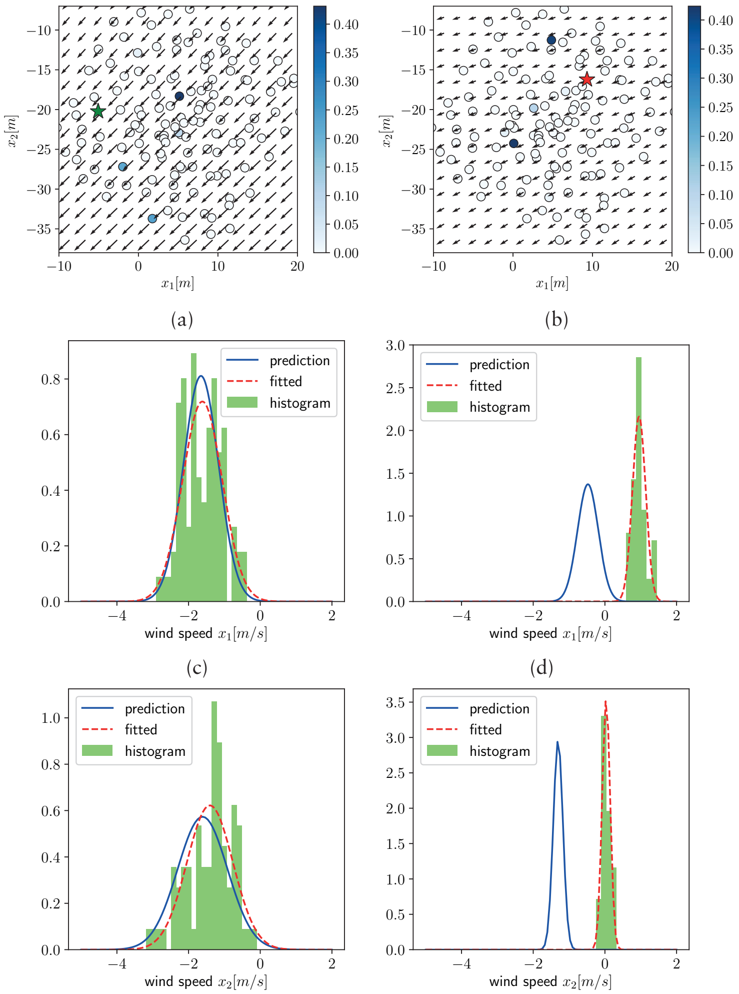


(a)

(b)

Figure 7.25: The figure shows the measured wind speed in (a) and wind direction (b) in an outdoor experiment over approximately 25 minutes. Further, the error in the model prediction is plotted (blue curves). The time stamps where the prediction is best and worst are marked by vertical lines.

In Figure 7.25 the worst case and the best case are marked by a red and green vertical line. Here "worst" and "best" is calculated according to the



(a) (b)

(c) (d)

(e) (f)

Figure 7.26: The plots on the left (c) and (e) compare the model prediction in the outdoor scenario with the actual measured wind data at the location indicated by a green star in (a), i.e. the best case. On the left (d) and (f) the same is shown for the location marked by a red star in (b), i.e. the worst case. Further, in (a) and (b) the whole predicted wind field is shown for the time stamps marked in Figure 7.25 ((a) green, (b) red). The circles in (a) and (b) indicate all measurement locations, and their color visualize their influence (weight) on the current prediction.

Kullback-Leibler divergences between the predicted Gaussian PDF and a fitted Gaussian to the actual measured data. These Gaussians are depicted in Figure 7.26 along with the histogram of the measured data (on the left: best case; on the right: worst case). Further, Figure 7.26a and 7.26b show the predicted instantaneous air flow field for the two cases. The stars indicate the best and worst evaluated location. In addition, the circles indicate all measured locations while their colors represent their weights, i.e. their influences on the prediction. Figure 7.25b makes it obvious that the worst case corresponds to a time when suddenly the wind direction changed. In case of such sudden changes, the model is not able to predict the air flow field accurately and needs some time and additional measurements to adapt.

Even though the wind conditions are difficult to predict, the model is able to estimate the wind speed with an average error of $0.5m/s$ and the wind direction with an error of less than $30deg$.

Let us finally remark on the parametrization of the air flow model for the outdoor scenario. In contrast to the indoor experiments in Section 7.3.2, the air flow field is not static in the outdoor experiments. Therefore, this time we have to optimize both kernel widths k_s and k_t according to Section 4.4. The two parameters are chosen so that the total Kullback-Leibler divergence from the fitted Gaussian to the predicted Gaussian is minimal. More precisely, all Kullback-Leibler divergences are summed up, i.e. for the two components of the air flow velocity and for all locations. This sum is minimized. For this optimization only the first half of the data set (approximately $12min$) is used, where the second half is used to terminate the optimization to avoid over-fitting. In our example the optimal parameters are $k_s = 345m^2$ and $k_t = 473s^2$, which corresponds to a spatial kernel width of approximately $19m$ and a temporal kernel width of $22s$.

7.3.3 Real-World Experiments

In this section the gas source localization strategy proposed in this thesis will be evaluated in experiments with “toy” gas.

The experiments were carried out in the same laboratory environment as the Hardware-in-the-Loop experiments in Section 7.3.1 and the indoor air flow mapping in Section 7.3.2 (see also Figure 7.14). The size of the considered environment is $3m$ times $6m$. The DARES approach uses FEM as a numerical approximation of gas dispersion (see Section 3.3.2) and the implementation as described in Section 5.2. Therefore, the environment is discretized by 16 times 31 nodes arranged in a grid. In the experiments, we consider the gas distribution as a static process, i.e. we consider the dispersion process in a steady state (at least within the time window the exploration takes place). Dirichlet boundary conditions are applied at the borders of the environment due to a lack of actual knowledge about the conditions. It is important to note that this assumption might be wrong and might entail an additional mismatch between

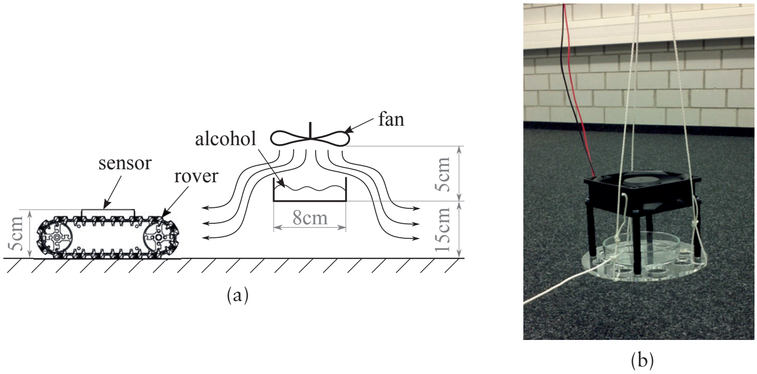
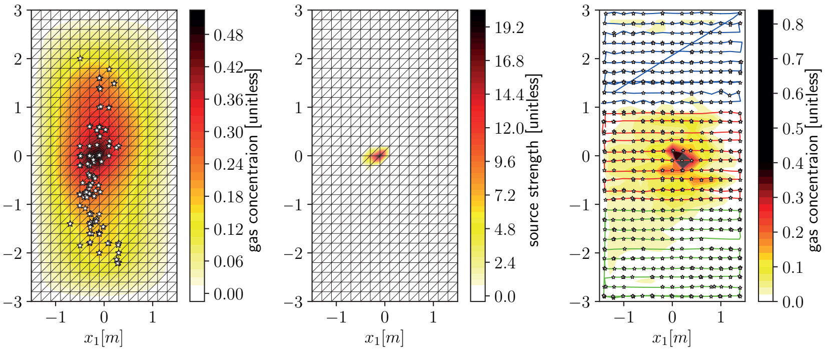


Figure 7.27: The schematic in (a) shows the setup of the real-world experiment. A culture dish containing ethanol as shown in (b) is hanging from the ceiling so that the robots do not collide with it. Above the dish a fan is mounted to support evaporation of the ethanol.

the reality and the used model. However, the proposed approach is designed to cope with such mismatch effects.

As a source, we used ethanol evaporating from a culture dish (8 cm diameter) filled with approximately 5g of 94% ethanol assay. Above the culture dish we mounted a small fan running at 5V (see Figure 7.27). The airflow caused by this fan avoids saturation of ethanol concentration in the layer above the liquid. Thus, it accelerates the evaporation. Moreover, the air flow facilitates a radial dispersion of the ethanol vapor. The whole structure hangs down from the ceiling, so that the robots are able to drive below the source without any collision with the dish containing the alcohol. It is also important that the culture dish is relatively small compared to the discretization of the environment (20 cm distance between nodes). In this way we make sure that no cell is completely below the dish and not totally covered by it, and that the highest concentration is still below the source.

In the following experiments three small robots as presented in Section 7.2.2 were employed. They make use of the DARES strategy of Section 6.2 and are coordinated according to the collision avoidance mechanism described in Section 6.3.1. Different experiments were performed, where the robots either carry MOX sensors (see Section 7.2.1) or PID sensors (see Section 7.2.1). All sensors were zeroed shortly before the experiments to account for offsets in the ADCs or background gas concentration in the room. In other words, sensors were placed far away from gas sources (approximately 4m) and were considered to be exposed to clean air. Then, the sensor parameters (see Section 7.2.1 and 7.2.1) were adjusted so that the sensors concentration measurements equal zero.



(a) (b) (c)

Figure 7.28: Real-world experiment 1: The figure shows the results of an experiment with a single source placed at $x_1 = 0\mathrm{m}$ and $x_2 = 0\mathrm{m}$. The DARES approach was parametrized as follows: 3 robots, $\kappa = 1.5$, $\tau_s = 10^2$, $\tau_m = 10^3$. In (a) the estimated gas concentration distribution is shown, while (b) depicts the estimated source distribution. In (c) the gas concentration distribution is shown which has been sampled by three robots following meander trajectories. The stars indicate measurement locations.

In the following, four representative experiments are presented and discussed. It is important to note that conducting real-world experiments is quite time consuming. Thus, the results presented here do not provide studies of parameters or effects with statistical significance as the simulations in Section 7.1 did. Instead, the purpose of the following experiments is to demonstrate that the source localization strategy works in real-world scenarios under controlled conditions as well as to shown possible limitations of the proposed scheme.

In the first experiments the robots were equipped with MOX sensors. At each measurement location the sensor readings were averaged over 5 seconds. A single source was placed in the middle of the exploration environment at $x_1 = 0\mathrm{m}$ and $x_2 = 0\mathrm{m}$ and no artificial air flow was introduced. Figure 7.28a and Figure 7.28b show the estimated gas concentration distribution and the estimated source distribution after approximately 360 seconds. Unfortunately, real-world experiments lack of ground truth data to accurately evaluate the performance of the source localization strategy. We only know the ground truth position of the source but not its strength. Therefore, we normalized the estimated gas source distribution by its maximum and compared it to a ground truth gas distribution with a single peak at $x_1 = 0\mathrm{m}$ and $x_2 = 0\mathrm{m}$ with strength 1. In this way we computed the EMD between the ground truth and the esti-

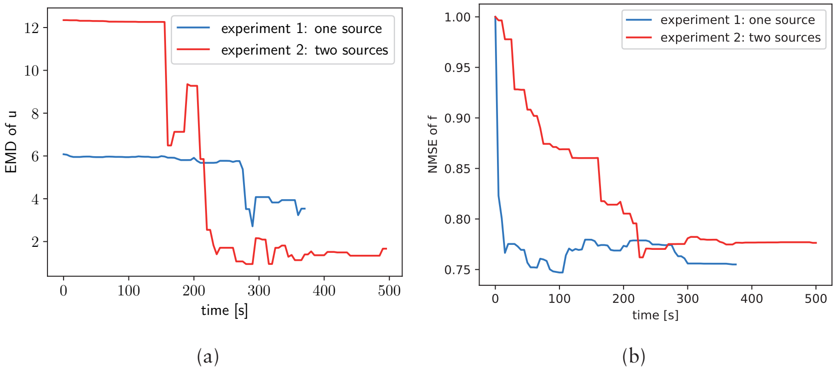


Figure 7.29: The figure shows the performance of the experiments corresponding to Figure 7.28 (one source) and to Figure 7.30 (two sources). The performance in measured by means of the EMD of the estimated source distribution in (a) and by means of the NMSE of the estimated gas concentration.

mated source distribution for each time stamp. The EMD for this first experiment is shown in Figure 7.29a. Also, a subjective impression obtained from the plotted source distribution in Figure 7.28b shows that the source is correctly localized. However, there are small imperfections. For example the estimated source distribution is not a single peak, but actually two peaks close to each other, located at $x_1 = -0.1\mathrm{m}, x_2 = 0\mathrm{m}$ and $x_1 = -0.3\mathrm{m}, x_2 = 0\mathrm{m}$. Further note: As can be seen in Figure 7.29a, the EMD drops approximately after 280 seconds, this time stamp may be considered as the time when the source distribution is successfully identified.

Due to the lack of ground truth gas concentration data, it is hard to evaluate the estimated gas concentration distribution that is shown in Figure 7.28a. Nevertheless, to generate a kind of ground truth, the gas distribution was uniformly sampled by meander trajectories as shown in Figure 7.28c. We compare the estimated gas concentration in Figure 7.28a to these samples. More precisely, we calculate the NMSE between the estimate $\vec{f}$ and the gas measurements sampled by meander trajectories. (Note that the samples and the $\vec{f}$ are first normalized by their L_2 norm.) The resulting error curve is shown in Figure 7.29b. The discrepancy is quite high and also a visual comparison between Figure 7.28a and Figure 7.28c shows that the two distributions do not match very well. This indicates a certain mismatch between the model and reality. The mismatch is not unexpected, since the model is a simplification and approximation of the true dispersion process. Motivated by this fact, the trust in our model was lowered in the experiments by reducing the precision τ_s compared to the simulations. It

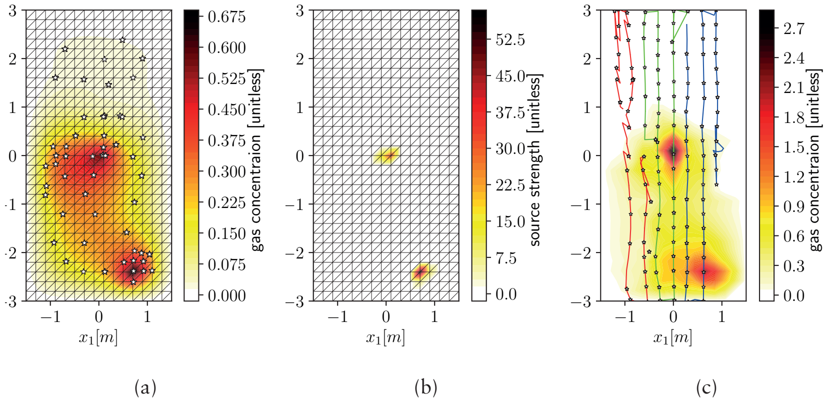
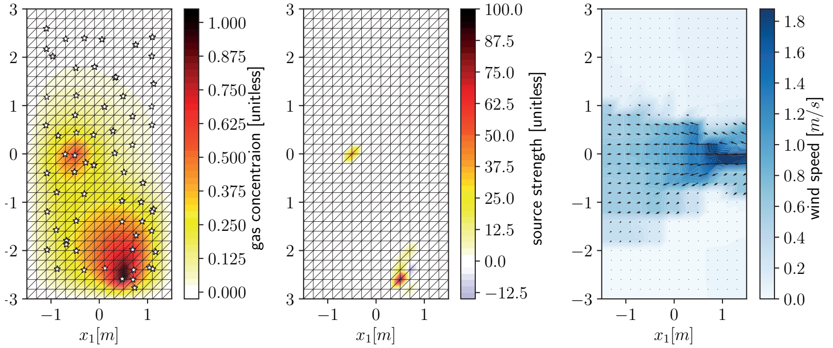


Figure 7.30: Real-world experiment 2: The figure shows the results of an experiment with two sources. The DARES approach was parametrized as follows: 3 robots, $\kappa = 1.5$, $\tau_s = 10^2$, $\tau_m = 10^3$. In (a) the estimated gas concentration distribution is shown, while (b) depicts the estimated source distribution. In (c) the gas concentration distribution is shown which has been sampled by three robots following meander trajectories. The stars indicate measurement locations.

is remarkable that, despite the mismatch of the predicted gas distribution, the source distribution is nearly perfectly estimated.

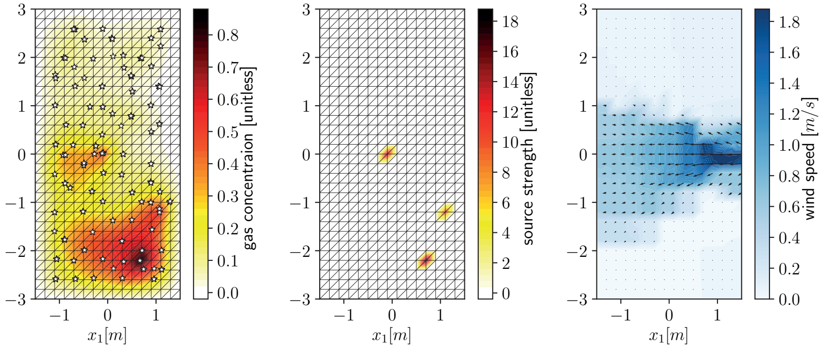
At this point let us remark that the MOX sensors performed very well in these kinds of experiments. However, they are not very convenient, since they had to be re-calibrated regularly (nearly every day). Thus, in the following experiments PID sensors were mounted to the robots. The PIDs are much more long-term stable. Thus, apart from zeroing before the experiment, they required no recalibration at all during the period in which the experimental studies were carried out (approximately two years).

The second experiment presented here is similar to the first one but with two sources, located at $x_1 = 0m$, $x_2 = 0m$ and $x_1 = 0.5m$, $x_2 = -2.4m$. Again the dispersion was not affected by air flow. Figure 7.30a and Figure 7.30b show the estimated gas concentration distribution and the estimated source distribution after approximately 500 seconds. Again the performance of the strategy and source estimation are depicted in Figure 7.29. The EMD and NMSE are calculated in the same way as explained of the first experiment above. For the ground truth source distribution it is assumed that the two sources have the same strength. This time the EMD drops significantly two times: once at approximately 150 seconds and once at 200 seconds. These times can be con-



(a) (b) (c)

Figure 7.31: Real-world experiment 3: The figure shows the estimated gas concentration in (a) and the estimated source distribution in (b) of an experiment with two sources and an artificial air flow field as depicted in (c). The DARES approach was parametrized as follows: 3 robots $\kappa = 1.5$, $\tau_s = 10^2$, $\tau_m = 10^3$. The stars indicate measurement locations.



(a) (b) (c)

Figure 7.32: Real-world experiment 4: The figure shows the estimated gas concentration in (a) and the estimated source distribution in (b) of an experiment with two sources and an artificial air flow field as depicted in (c). The DARES approach was parametrized as follows: 3 robots $\kappa = 0.5$, $\tau_s = 10^2$, $\tau_m = 10^3$. The stars indicate measurement locations. In this experiment apart from the two sources at $x_1 = 0\text{m}$, $x_2 = 0\text{m}$ and $x_1 = 0.5\text{m}$, $x_2 = -2.4\text{m}$ a third source is localized erroneously at $x_1 = 1.1\text{m}$, $x_2 = -1.2\text{m}$.

sidered as the times when the two sources are successfully localized. Also, the visual impression in Figure 7.30b indicates that the source distribution is estimated quite well.

For the third and fourth experiment an artificial air flow was generated by five fans as described in Section 7.3.2. DARES incorporates the probabilistic air flow map that was developed in Section 4.4 and parametrized as discussed in Section 7.3.2. This air flow map is also depicted in Figure 7.31c and Figure 7.32c. Apart from the air flow map the experimental setup was the same as for the second experiment. Two sources were placed in the laboratory (at $x_1 = 0\mathrm{m}$, $x_2 = 0\mathrm{m}$ and $x_1 = 0.5\mathrm{m}$, $x_2 = -2.4\mathrm{m}$). The performance measured by means of the EMD is plotted in Figure 7.33.

Let us have a closer look at the results: As shown in Figure 7.31, in the third experiment the estimated source distribution correctly contains two peaks corresponding to the two sources. However, the source placed at $x_1 = 0\mathrm{m}$, $x_2 = 0\mathrm{m}$ is estimated at location $x_1 = -0.5\mathrm{m}$, $x_2 = 0\mathrm{m}$, which is $0.5\mathrm{m}$ downstream of the true position.

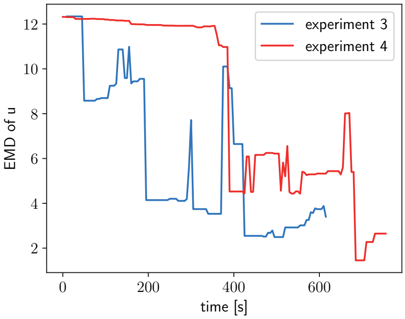


Figure 7.33: The plot shows the performance of the experiments corresponding to Figure 7.31 and to Figure 7.32. The performance in measured by means of the EMD of the estimated source distribution.

In the fourth experiment again the two sources are correctly estimated indicated by the two corresponding peaks in the estimated source distribution in Figure 7.32. In addition, the estimated source distribution contains a third peak at $x_1 = 1.1\mathrm{m}$, $x_2 = -1.2\mathrm{m}$ that is wrong and does not correspond to any source. The reason for this wrong estimate is most likely a wrong boundary condition. The boundary condition claims that the gas concentration at the border is zero. Unfortunately, in this experiment the background gas concentration in the room was not zero but higher, as can be seen by the gas distribution depicted in Figure 7.32a (especially close to the $x_1 = 1.1\mathrm{m}$, $x_2 = -1.2\mathrm{m}$). A sudden change in the gas concentration towards zero at the border can only be explained by the gas dispersion model if there would be a source. Thus, the model placed erroneously a third source in this region.

To sum up, the proposed DARES strategy could also be demonstrated to work in real-world experiments with and without airflow. The mismatch between model assumptions and the real gas dispersion deteriorates the performance compared to the simulation studies a little bit. Nevertheless, qualitatively speaking, the peaks in the estimated source distribution are very close to the true source positions.

Chapter 8
Conclusion

This final chapter will summarize the thesis and its contributions, and relate them to the big picture. We will also discuss advantages and limitations of the presented work as well as open questions and possible future research topics resulting from this thesis.

8.1 Summary and Contributions

To sum up the contribution of the whole thesis in a single statement: The thesis shows how to support robotic exploration tasks by a-priori available domain knowledge and reveals the advantages of doing so. In particular, it is shown by means of gas source localization. As a solution, we propose the Domain-knowledge Assisted Robotic Exploration and Source-localization (DARES) approach. There, we encode domain knowledge about the physical phenomenon of gas dispersion in a mathematical model that supports a multi-robot system to find several gas sources.

As a suitable mathematical model, we identify the advection-diffusion PDE in Chapter 3. The PDE approximates the gas dispersion process accurately enough, while the numerical complexity of the PDE is still tractable in real-time robotic applications. In the end as a result of Chapter 3, our gas dispersion model becomes a system of algebraic equations. With the use of this model two main advantages appear. First the model allows us to infer the source distribution based on gas concentration measurements taken by robots; this is an estimation problem. Second the model enables us to quantify the uncertainty of this estimated source distribution. The uncertainty is used later on in Chapter 6 to design an exploration strategy that is guiding the robots to informative measurement locations.

The thesis proposes to embed the gas dispersion model into a probabilistic Bayesian framework as done in Chapter 4. The probabilistic framework not only facilitates the uncertainty quantification, but also allows to relax model assumptions. To be precise, we do not consider the residuals of the algebraic equation system to be exactly equal zero. Instead, we model residuals as a stochastic process that is Gaussian distributed with mean zero. The variance of the Gaussian distribution is used to parametrize our trust into our model. To facilitate this possibility is one of the main contributions of the thesis.

Moreover, in the Bayesian approach we can naturally incorporate additional a-priori domain knowledge by means of appropriate prior distributions. This is very useful, since in the thesis we consider the number of sources as unknown but sparsely distributed in the environment. By means of SBL techniques, Chapter 4 shows how to design a prior PDF to model the assumption that the sources are sparsely distributed. This turned out to be another important contribution of the thesis. The sparsity inducing prior first of all acts as regularization mechanism for the source estimation problem. Therefore, it is possible to estimate or approximate a source distribution even from only few gas concentration measurements. Without the sparsity inducing prior the estimation problem would be underdetermined and not solvable in the early phase of the exploration, when only an insufficient number of measurements is available. Second, simulations and experiments have shown that the sparsity inducing prior is actually responsible for guiding the robots to the informative location close to sources.

Based on the probabilistic model, the design of our DARES strategy in Chapter 6 is straightforward. The probabilistic model provides a marginal PDFs for the estimated gas concentration for individual regions in the environment. According to information theory a high variance of the PDF corresponds to high entropy which can be considered as a high uncertainty in the estimated gas distribution. The DARES strategy therefore sends the robots to locations where the variance in the estimated gas concentration is high. To this end Chapter 6 also proposes mechanisms to coordinate the multi-robot system and to decide which robot is going where.

The presented DARES strategy is designed for a multi-robot system. Multiple robots are not only able to take measurements at the same time at different locations, which is important to observe a dynamic process; they can also make use of synergies and share computational costs. In this respect, Chapter 5 contributes a novel approach to carry out necessary calculations required by the DARES strategy in a distributed fashion on on-board computers of the robots. This approach is based on a factor graph representation of the probabilistic model and message passing.

All individual contributions are combined and evaluated in Chapter 7 by means of simulations, Hardware-in-the-Loop experiments and experiments with ethanol.

The thesis was guided by four research questions which have been introduced in Section 1.3. In the following we provide comprehensive answers to these questions that are also summarizing the contribution of the thesis again.

? **How to incorporate domain knowledge from physics into an exploration strategy for gas source localization?**

✓ PDEs encode physical knowledge about dynamic processes in a mathematical model. Such a model can be used to design an uncertainty-driven exploration strategy. The strategy guides robots to informative measurement locations that facilitate estimating the gas sources based on concentration measurements.

? **How can we quantify the trust into our model and take it into account in the exploration strategy?**

✓ By translating the model used by the exploration strategy into a probabilistic framework, we can allow residuals of the model to deviate from zero according to a probability distribution. By the precision (or variance) of the distribution we can parametrize the trust into our model.

? **How can we avoid assumptions regarding the number of sources?**

✓ In our Bayesian approach we can design a sparsity inducing prior for the spatial source distribution which we are looking for. In this way the estimated source distribution contains only a few sources without knowing the exact number of sources a-priori.

? **How to take advantage of the multi-robot system in the implementation?**

✓ Marginal PDFs required by the exploration strategy can be calculated by a message passing algorithm. Such an algorithm can be implemented in a distributed fashion so that the robots split the computational cost among themselves.

8.2 Discussion and Limitations

At this point let us first consider from a general point of view the idea of assisting a robotic system by available domain knowledge. While in general it seems reasonable, nowadays there is a tendency to make use of artificial intelligent methods where robotic systems learn to understand their environment on their own based on data or thousands of experiments of trial and error. However, why should a robot learn something the hard way that scientists and engineers already know? The answer lies in the fact that it is often difficult to provide human knowledge to a robotic system. In this thesis we found a way to encode available knowledge about gas dispersion from physics in a mathematical model based on a PDE. This mathematical model is suitable for an implementation in a robotic system. However, in general to find such a model for the process of interest includes a lot of work or might be too complex for some physical phenomenons.

As we would expect, the additional information on the physical phenomenon of gas dispersion helps the robots to localize the sources. The robots start to sample the gas concentration in downwind regions which also a human expert would confirm as most informative. Further, as soon as a high gas concentration is measured indicating the present of a source, the robots concentrate their measurements around this location. Of course this behavior is very beneficial to localize the sources accurately and with as few measurements as possible. However, it is difficult to understand where in the mathematical formalism this behavior is coming from. Here we would like to offer an attempt at an intuitive explanation: As we have seen in Section 7.1.2, the behavior arises from the DARES strategy in combination with the sparsity inducing prior PDF. The prior PDF encodes the assumption that the sources are sparsely distributed in the environment. In other words, the prior assumes that for most regions (cells or nodes) the source strength is zero. Under this assumption, evidence for a source with non-zero posterior mean effectively "contradicts" this prior assumption; as such the uncertainties in the corresponding regions grow. The increased uncertainty attracts the robots that are guided by DARES and thus the robots concentrate their measurements on informative regions around the sources. This effect also has a downside. Favoring the regions around the sources, may cause single robots to get stuck in the neighborhood of a source. This may prevent — under certain conditions and an insufficient number of robots — to discover all sources. Multiple robots, however, are able to effectively reduce the uncertainty around the sources by multiple simultaneous measurements. Thus, individual robots can "escape" from the attraction of a source location and can explore other regions. This fact also shows the advantage of employing multiple robots for gas source localization.

In the thesis the source localization problem is formulated as the problem of estimating a spatial source distribution. The source distribution can represent any number of sources only limited by the spatial discretization. In this way the approach does not require knowing the exact number of sources beforehand. However, representing the sources by a distribution has also disadvantages. While a single source may be described by a 2D vector containing its spatial coordinates, the source distribution used in this thesis is represented by a high dimensional vector containing the source strength at all discrete cells. Estimating the high dimensional representation of the source distribution is difficult even when supported by the sparsity inducing prior. Especially in larger real-world scenarios, more complex than the laboratory environment considered here, the estimated source distribution based on few noisy gas measurements may be of poor quality. Many measurements would be required to estimate the whole distribution accurately. In contrast by a different approach a single source might be estimated with much fewer measurements. So, the here presented approach should not be applied if the number of sources is a-priori known.

A further limitation of the presented approach in complex real-world applications may be that accurate gas concentration measurements are required. Measuring absolute gas concentrations accurately is complicated by common sensor characteristic as noise, slow response and recovery times, cross-sensitivity and interference by other environmental parameters, e.g. humidity. In addition, a patchy and turbulent structure of the gas distribution makes meaningful measurements difficult and would require averaging measurements over long times. On the other side, long and complicated measurement procedures also motivate using the proposed approach. If measurements are complicated or expensive (e.g. with respect to time), it is important to localize the sources with as few measurements as possible.

8.3 Future Topics

The work of the thesis also leads to a number of new questions that may be addressed in the future. Here we summarize the most important ones.

First, while for the linear dispersion model the thesis proposes a message passing algorithm to calculate the posterior PDF in a distributed fashion (Section 5.1), for the non-linear model only a centralized implementation is given in Section 5.2. The modification of this algorithm to be suitable for a distributed implementation on a multi-robot system is part of future work. In fact, the mean field approximation used in Section 5.2 could be applied more extensively to have an even stronger factorized form in equation (5.33) (namely to have a factorized version of $q_\theta(\vec{\theta})$). Then the algorithm would require updating many individual factors which can be done in parallel by the robots. (Note that even now the update of $q_{\gamma_j}(\gamma_i)$ can be done in parallel.)

Second, the assignment of way points to robots and coordinating the robots in Chapter 6 was designed more with respect to robustness rather than optimality. Here, future research efforts may lead to more efficient approaches. It is important to note that the coordination of robots is very crucial for the overall performance. Informative way points proposed by DARES are useless, if the robots cannot reach them because their way is blocked by others. In this respect also Artificial Intelligence methods are considerable in the future. For example, a Reinforcement Learning approach for information gathering similar to [184], may be applied for gas source localization. The robots can learn an optimal movement strategy by rewarding them for reducing the uncertainty comparable to the proposed DARES strategy.

Since assisting the exploration by domain knowledge has shown great potential, it would be interesting to see a similar approach for air flow modeling. While in the thesis the air flow model was obtained purely data driven, its estimation can also incorporate knowledge from physics. In fact, similar to the gas dispersion model of this thesis based on the advection-diffusion PDE, an air flow model might be estimated based on domain knowledge encoded by the Navier-Stokes PDEs (or simplified cases of them). In general, it is worth looking at the air flow model, and elaborating and improving it, since wind is the dominating mechanism of gas dispersion.

Also, the probabilistic Bayesian framework may offer a playground for further research. For example, additional or other prior PDFs may be developed to incorporate additional assumptions. For instance, a possible prior may restrict source strengths and gas concentrations to positive values only. So far negative sources and negative gas concentrations could occur in the estimates from the mathematical point of view. While a negative source strength might be considered as a kind of sink, a negative concentration has no physical meaning.

Even though, the thesis studied a lot of different effects like the sparsity inducing prior, model mismatch effects etc, further evaluations in the future are possible. For example, how does the proposed DARES strategy perform in case of many sources that are no longer sparsely distributed. Another interesting question would be how the diffusion parameter κ influence the exploration performance. In the thesis this diffusion parameter κ was chosen imperially. In the future the presented approach might be extended so that the diffusion parameter is also estimated based on collected measurements. Further, in a next step the proposed approach should be tested in more realistic and more complex outdoor scenarios.

Last but not least, a future research direction might extend the proposed framework presented in the thesis to remote sensing and gas tomography. In this respect, the main challenge would be to adapt the sensor model to integral gas concentration measurements as provided by open path TDLAS sensors. Further, for efficient source localization the sampling strategy would be more involved. The robots need to be positioned and orientated optimally (similar to [20]) such that the measurement beams are most informative, i.e. the uncertainty is reduced most.

Appendix A
Derivation of Advection Term

The following provides a detailed derivation of the advection term in the numerical approximation of the PDE based on FEM in Section 3.3.2. In the thesis this derivation is summarized in equation (3.31). We consider the following approximation:

$$\begin{aligned}
&\int_\Omega \varphi(\vec{x},t)\vec{v}(\vec{x},t)^\mathsf{T}\nabla f(\vec{x},t)\mathrm{d}\vec{x} \approx && \text{(A.1)}\\
&\approx \int_\Omega \left(\sum_j [\vec{\varphi}]_j(t)\psi_j(\vec{x})\right)\left(\sum_k [\vec{v}_1]_k(t)\hat{\psi}_k(\vec{x})\right)\left(\sum_i [\vec{f}]_i(t)[\nabla\psi_i(\vec{x})]_1\right)\mathrm{d}\vec{x}+ \\
&+\int_\Omega \left(\sum_j [\vec{\varphi}]_j(t)\psi_j(\vec{x})\right)\left(\sum_k [\vec{v}_2]_k(t)\hat{\psi}_k(\vec{x})\right)\left(\sum_i [\vec{f}]_i(t)[\nabla\psi_i(\vec{x})]_2\right)\mathrm{d}\vec{x}.
\end{aligned}$$

Since the following derivation is the same for both addends in equation (A.1) (first addend = second line, second addend = third line), we show the derivation only for the first addend. Let's rearrange the elements in the integral:

$$\begin{aligned}
&\int_\Omega \left(\sum_j [\vec{\varphi}]_j(t)\psi_j(\vec{x})\right)\left(\sum_k [\vec{v}_1]_k(t)\hat{\psi}_k(\vec{x})\right)\left(\sum_i [\vec{f}]_i(t)[\nabla\psi_i(\vec{x})]_1\right)\mathrm{d}\vec{x} = \\
& && \text{(A.2)}\\
&= \int_\Omega \left(\sum_j\sum_k [\vec{\varphi}]_j(t)[\vec{v}_1]_k(t)\psi_j(\vec{x})\hat{\psi}_k(\vec{x})\right)\left(\sum_i [\vec{f}]_i(t)[\nabla\psi_i(\vec{x})]_1\right)\mathrm{d}\vec{x}.
\end{aligned}$$

Since the basis functions $\psi_j(\vec{x})$ are Lagrange elements of order one (i.e. linear on the element j) and $\hat{\psi}_k(\vec{x})$ are Lagrange elements of order zero (i.e. constant 1 on the element k), we can assume:

$$\begin{aligned} \psi_j(\vec{x})\hat{\psi}_k(\vec{x}) &= \psi_j(\vec{x}) \quad \text{if} \quad j = k \\ \psi_j(\vec{x})\hat{\psi}_k(\vec{x}) &= 0 \quad \text{if} \quad j \neq k. \end{aligned} \tag{A.3}$$

This results in:

$$\begin{aligned} &\int_\Omega \left(\sum_j [\vec{\phi}]_j(t)[\vec{v}_1]_j(t)\psi_j(\vec{x}) \right) \left(\sum_i [\vec{f}]_i(t)[\nabla\psi_i(\vec{x})]_1 \right) d\vec{x} = \\ &\int_\Omega \sum_j \sum_i [\vec{\phi}]_j(t)[\vec{v}_1]_j(t)\psi_j(\vec{x})[\vec{f}]_i(t)[\nabla\psi_i(\vec{x})]_1 d\vec{x} \end{aligned} \tag{A.4}$$

In matrix form this can be written as:

$$\vec{\phi}(t)^\mathsf{T}\vec{v}_1(t) \circ \tilde{\mathbf{G}}_1\vec{f}(t), \tag{A.5}$$

with use of the Hadamard product $\circ$ and with

$$[\tilde{\mathbf{G}}_1]_{j,i} = \int_\Omega \psi_j(\vec{x})[\nabla\psi_i(\vec{x})]_1 d\vec{x}. \tag{A.6}$$

Proof:

$$\begin{aligned} &\vec{\phi}(t)^\mathsf{T}\vec{v}_1(t) \circ \tilde{\mathbf{G}}_1\vec{f}(t) = \\ &= \vec{\phi}(t)^\mathsf{T}\mathbf{A}\vec{f}(t); \qquad \text{with} \quad [\mathbf{A}]_{j,i} = [\vec{v}_1]_j(t)[\tilde{\mathbf{G}}_1]_{j,i} \end{aligned}$$

$$\begin{aligned} &\vec{\phi}(t)^\mathsf{T}\mathbf{A}\vec{f}(t) = \\ &\vec{\phi}(t)^\mathsf{T}\vec{b}; \qquad \text{with} \quad [\vec{b}]_j = \sum_i [\mathbf{A}]_{j,i}[\vec{f}]_i(t)] = \sum_i [\vec{v}_1]_j(t)[\tilde{\mathbf{G}}_1]_{j,i}[\vec{f}]_i(t) \end{aligned}$$

$$\vec{\phi}(t)^\mathsf{T}\vec{b} = c; \qquad \text{with} \quad c = \sum_j [\vec{\phi}]_j(t)[\vec{b}]_j = \sum_j \sum_i [\vec{\phi}]_j(t)[\vec{v}_1]_j(t)[\tilde{\mathbf{G}}_1]_{j,i}[\vec{f}]_i(t)$$

with equation (A.6) we finally get:

$$\begin{aligned} &\vec{\phi}(t)^\mathsf{T}\vec{v}_1(t) \circ \tilde{\mathbf{G}}_1\vec{f}(t) = \\ &= \int_\Omega \sum_j \sum_i [\vec{\phi}]_j(t)[\vec{v}_1]_j(t)\psi_j(\vec{x})[\vec{f}]_i(t)[\nabla\psi_i(\vec{x})]_1 d\vec{x} \end{aligned}$$

□

Appendix B
Collision Avoidance

Figure B.1 depicts the flow chart of the collision avoidance mechanism introduced in Section 6.3.2. The flow chart shows a state machine with five states: driving, collision impending, waiting, re-planning, carefully driving. The blocks implement the following functionality:

- **check for collision:** add all robots with a fewer distance than the safety distance to the collision list; returns true, if a new one occurred, otherwise false. Robots with a higher distance than the safety distance plus a hysteresis are removed from the list
- **check for someone with higher priority:** returns true, if any robot in the collision list has higher priority than oneself; otherwise false
- **waited for x sec:** checks, if the current state (i.e. driving, collision impending, waiting, re-planning or carefully driving) was entered more than x seconds ago
- **nobody with higher prior is moving:** returns true, if nobody in the collision list with higher priority is moving
- **cancel current goal (and store it):** sends a cancel command to the path planner; thus the robot stops; current goal is buffered in a local variable
- **publish stored goal:** sends the buffered goal to path planner; thus the robot starts to move.

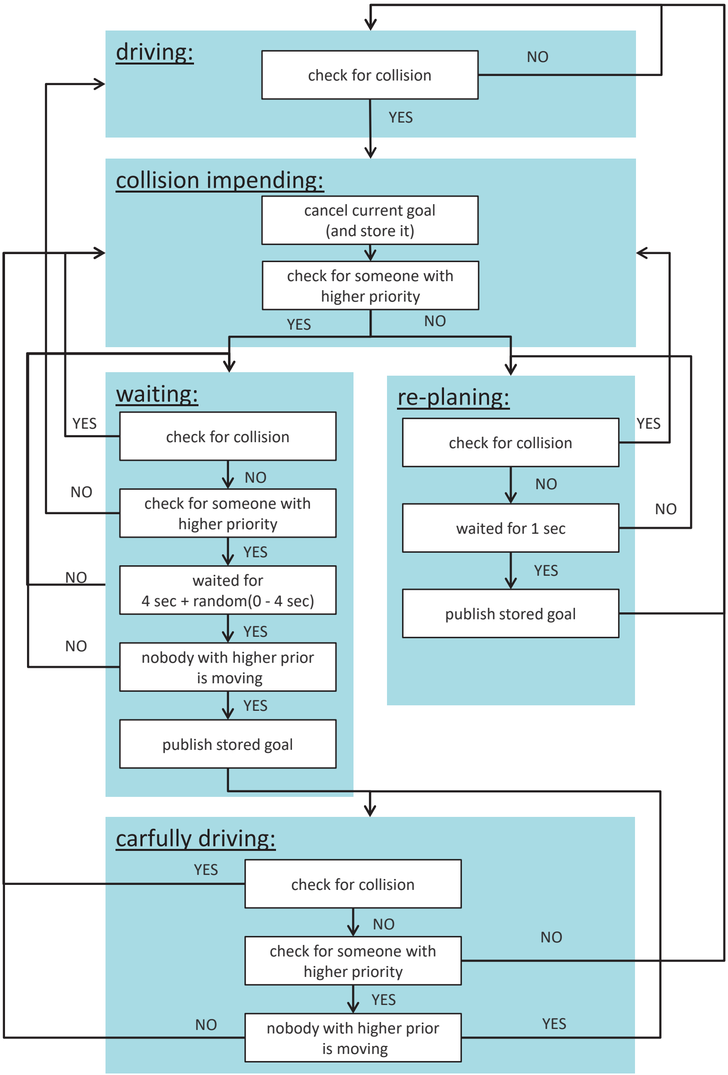

driving:
check for collision
NO
YES
collision impending:
cancel current goal (and store it)
check for someone with higher priority
YES
NO
waiting:
YES
check for collision
NO
NO
check for someone with higher priority
YES
NO
waited for 4 sec + random(0 - 4 sec)
YES
NO
nobody with higher prior is moving
YES
publish stored goal
re-planing:
check for collision
YES
NO
NO
waited for 1 sec
YES
publish stored goal
carfully driving:
YES
check for collision
NO
check for someone with higher priority
NO
YES
NO
nobody with higher prior is moving
YES

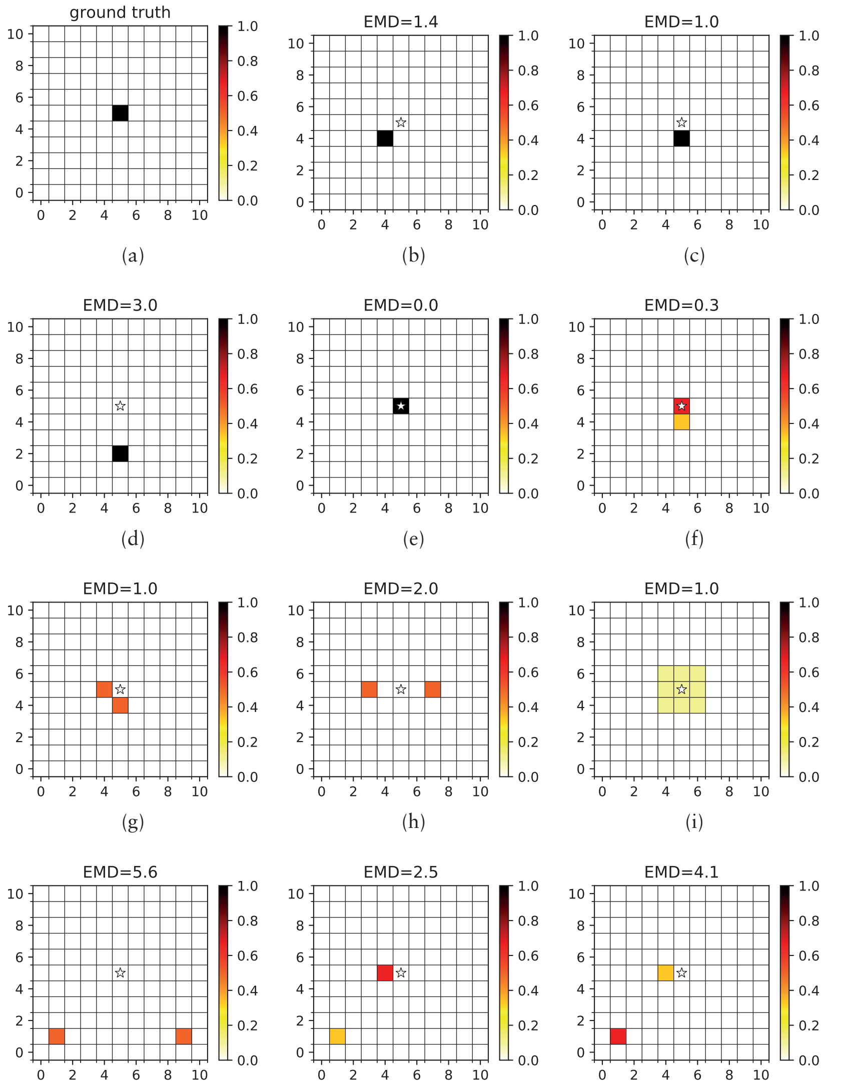
Figure B.1: The Figure depicts a detailed flow chart of the collision avoidance mechanism introduced in Section 6.3.2.

Appendix C
Error Metrics

C.1 Earth Mover's Distance (EMD)

The Earth Mover's Distance (EMD) measures the amount of work to turn one distribution into another distribution [151]. Vividly speaking, the EMD considers both distributions as piles of mud (earth). It measures the minimum effort to move the mass of the first distribution into the second distribution, where the effort is the amount of mass times the distance by which the mass has to be moved. In general, it is applied to discretized PDFs (histograms). In this case the EMD is a true measure of distance, since both distributions are normalize (have the same integral). A lower EMD indicates a higher similarity between the distributions. It can be considered as the discrete equivalent of the first Wasserstein metric. In order to calculate the EMD, the movement of mass can be considered as a transportation problem from linear optimization [151], where the first distribution plays the role of suppliers and the second distribution the role of consumers.

In this thesis the EMD is used to measure the distance between sparse source distributions, namely the estimated source distribution and its ground truth. Before applying the EMD, the source distributions are also normalized so that both contain the same amount of mass. The EMD is very suitable to measure the distance between sparse distributions as in our case. For example in case the source distribution contains a single peak that is estimated correctly but shifted by a grid cell, the EMD accounts for the distance of the shift. In contrast, the NMSE would not account for this effect and the error would be the same if the peak is shifted by a single cell or multiple cells. Unfortunately, the actual value of the EMD is not very intuitive. Therefore, Figure C.1 and C.2 show some examples of distributions and their EMD compared to a ground truth distribution depicted in Figure C.1a and Figure C.2a. Figure C.1 can be considered as the case of a single source at $x_1 = 5, x_2 = 5$ and Figure C.2 as two sources located at $x_1 = 3, x_2 = 5$ and $x_1 = 7, x_2 = 5$ (marked by starts).



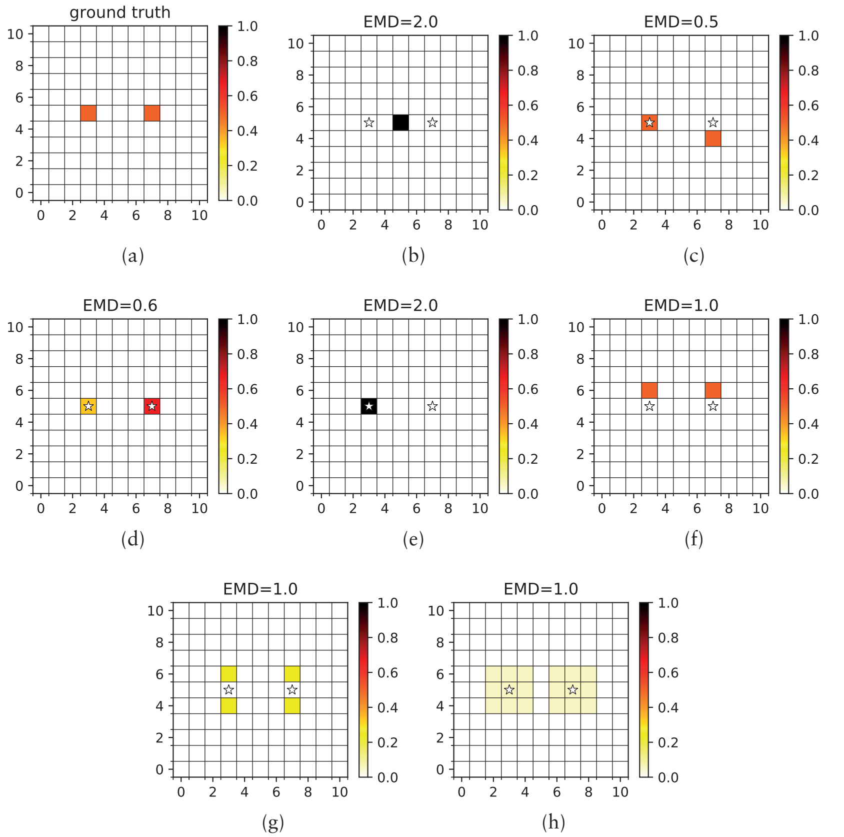
(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Figure C.1: This figure shows spatial distributions in the x_1-x_2 plane. The distribution in (a) is considers as the ground truth. All other distributions are compared to the ground truth by means of the EMDs.



(a) (b) (c)

(d) (e) (f)

(g) (h)

Figure C.2: This figure shows spatial distributions in the x_1-x_2 plane. The distribution in (a) is considers as the ground truth. All other distributions are compared to the ground truth by means of the EMDs.

C.2 Normalized Mean Square Error (NMSE)

During the thesis the NMSE is used to measure how well the gas concentration distribution is estimated. It compares the estimated concentration represented by $\vec{f}_{est}[n]$ to a ground truth gas concentration $\vec{f}_{gt}[n]$ according to the following equation:

$$\mathrm{NMSE} = \frac{\left\|\vec{f}_{gt}[n] - \vec{f}_{est}[n]\right\|^2}{\left\|\vec{f}_{gt}[n]\right\|^2} \tag{C.1}$$

References

[1] MiCS-5524 Data Sheet. Technical report, SGX Sensortech. (Cited on pages 111 and 112.)

[2] Numpy: https://numpy.org version 1.18.1. (Cited on pages 50 and 77.)

[3] Robot Operating System (ROS) http://www.ros.org/ (Date Accessed: 2019-11-28). (Cited on page 114.)

[4] Scikit-learn: https://scikit-learn.org version 0.22.1. (Cited on pages 50 and 77.)

[5] SciPy: https://www.scipy.org/ version 1.4.1. (Cited on page 81.)

[6] Ultrasonic Anemometer Model 86000. Technical Report 86000-90(E), R.M. Young Company. (Cited on page 113.)

[7] www.au.dk/oml-international (Date Accessed: 2020-01-07). (Cited on page 27.)

[8] www.figarosensor.com/technicalinfo/ (Date Accessed: 2019-12-02). (Cited on pages 13 and 14.)

[9] www.smprobotics.com/products_autonomous_ugv/gas-leak-inspection-robot/ (Date Accessed: 2020-04-09). (Cited on page 18.)

[10] PID-AH2 Photo Ionisation Detector aata sheet. Technical report, Alphasense Limited, 2017. (Cited on pages 112 and 113.)

[11] John. D. Albertson, Tierney Harvey, Greg Foderaro, Pingping Zhu, Xiaochi Zhou, Silvia Ferrari, M. Shahrooz Amin, Mark Modrak, Halley Brantley, and Eben D. Thoma. A Mobile Sensing Approach for Regional Surveillance of Fugitive Methane Emissions in Oil and Gas Production. *Environmental Science & Technology*, 50(5):2487–2497, 2016. (Cited on pages 7 and 87.)

[12] Alen Alexanderian, Noemi Petra, Georg Stadler, and Omar Ghattas. A-optimal design of experiments for infinite-dimensional Bayesian linear inverse problems with regularized l0-sparsification. *SIAM Journal on Scientific Computing*, 36(5):A2122–A2148, 2014. (Cited on page 88.)

[13] Grant Allen, Peter Hollingsworth, Khristopher Kabbabe, Joseph R. Pitt, Mohammed I. Mead, Samuel Illingworth, Gareth Roberts, Mark Bourn, Dudley E. Shallcross, and Carl J. Percival. The development and trial of an unmanned aerial system for the measurement of methane flux from landfill and greenhouse gas emission hotspots. *Waste Management*, 87:883–892, 2019. (Cited on page 3.)

[14] Alphasense. Application Note AAN 301-04 Introduction to PhotoIonisation Detection. Technical report, Alphasense Limited, 2010. (Cited on pages 15, 16, and 112.)

[15] Alphasense. Application Note AAN 104 How Electrochemical Gas Sensors Work. Technical report, Alphasense Limited, 2013. (Cited on pages 13 and 14.)

[16] Alphasense. Application Note AAN 305-06 VOC Correction Factor. Technical report, Alphasense Limited, 2017. (Cited on page 112.)

[17] Miguel Alvarado, Felipe Gonzalez, Peter Erskine, David Cliff, and Darlene Heuff. A Methodology to Monitor Airborne PM10 Dust Particles Using a Small Unmanned Aerial Vehicle. *Sensors*, 17(2):343, 2017. (Cited on pages 19 and 22.)

[18] Miguel Alvarado, Felipe Gonzalez, Andrew Fletcher, and Ashray Doshi. Towards the development of a low cost airborne sensing system to monitor dust particles after blasting at open-pit mine sites. *Sensors*, 15(8):19667–19687, 2015. (Cited on page 18.)

[19] Joshua S. Apte, Kyle P. Messier, Shahzad Gani, Michael Brauer, Thomas W. Kirchstetter, Melissa M. Lunden, Julian D. Marshall, Christopher J. Portier, Roel C.H. Vermeulen, and Steven P. Hamburg. High-Resolution Air Pollution Mapping with Google Street View Cars: Exploiting Big Data. *Environmental Science & Technology*, 51(12):6999–7008, 2017. (Cited on pages 22 and 29.)

[20] Muhammad Asif Arain, Marcello Cirillo, Victor Hernandez Bennetts, Erik Schaffernicht, Marco Trincavelli, and Achim J Lilienthal. Efficient Measurement Planning for Remote Gas Sensing with Mobile Robots. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3428–3434, 2015. (Cited on pages 16, 18, 23, and 144.)

[21] Richard M. Arrieta, Wei Li, Jay A. Farrell, and Shuo Pang. Initial development and testing of an adaptive mission planner for a small unmanned underwater vehicle. In *Proceedings of the International Conference on Offshore Mechanics and Arctic Engineering (OMAE)*, pages 609–617, 2003. (Cited on page 17.)

[22] Sahar Asadi, Han Fan, Victor Hernandez Bennetts, and Achim J. Lilienthal. Time-dependent gas distribution modelling. *Robotics and Autonomous Systems*, 96:157–170, 2017. (Cited on pages 22 and 60.)

[23] Maurice Stevenson Bartlett. An Inverse Matrix Adjustment Arising in Discriminant Analysis. *The Annals of Mathematical Statistics*, 22(1):107–111, 1951. (Cited on page 69.)

[24] Behzad Bayat, Naveena Crasta, Alessandro Crespi, António M. Pascoal, and Auke Ijspeert. Environmental monitoring using autonomous vehicles: a survey of recent searching techniques. *Current Opinion in Biotechnology*, 45:76–84, 2017. (Cited on page 7.)

[25] Behzad Bayat, Naveena Crasta, Howard Li, and Auke Ijspeert. Optimal search strategies for pollutant source localization. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1801–1807, 2016. (Cited on page 87.)

[26] Victor Hernandez Bennetts, Kamarulzaman Kamarudin, Thomas Wiedemann, Tomasz Piotr Kucner, Sai Lokesh Somisetty, and Achim J. Lilienthal. Multi-domain airflow modeling and ventilation characterization using mobile robots, stationary sensors and machine learning. *Sensors*, 19(5), 2019. (Cited on page 59.)

[27] Victor Hernandez Bennetts, Tomasz Piotr Kucner, Erik Schaffernicht, Patrick P Neumann, Han Fan, and Achim J Lilienthal. Probabilistic Air Flow Modelling Using Turbulent and Laminar Characteristics for Ground and Aerial Robots. *IEEE Robotics and Automation Letters*, 2(2):1117–1123, 2017. (Cited on page 59.)

[28] Victor Hernandez Bennetts, Achim J. Lilienthal, Patrick P. Neumann, and Marco Trincavelli. Mobile robots for localizing gas emission sources on landfill sites: Is bio-inspiration the way to go? *Frontiers in Neuroengineering*, 4(JANUARY):1–12, 2012. (Cited on pages 3, 15, 17, and 21.)

[29] Victor Hernandez Bennetts, Achim J. Lilienthal, and Marco Trincavelli. Creating true gas concentration maps in presence of multiple heterogeneous gas sources. In *Proceedings of IEEE Sensors*, pages 1–4, 2012. (Cited on pages 16 and 18.)

[30] Victor H.ernandez Bennetts, Erik Schaffernicht, Victor Pomareda, Achim J. Lilienthal, Santiago Marco, and Marco Trincavelli. Combining non selective gas sensors on a mobile robot for identification and mapping of multiple chemical compounds. *Sensors*, 14(9):17331–17352, 2014. (Cited on page 22.)

[31] Victor Hernandez Bennetts, Erik Schaffernicht, Todor Stoyanov, Achim J. Lilienthal, and Marco Trincavelli. Robot assisted gas tomography - Localizing methane leaks in outdoor environments. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 6362–6367. IEEE, 2014. (Cited on pages 16, 17, 18, and 23.)

[32] Christopher M Bishop and Michael E Tipping. Variational Relevance Vector Machines. In *Proceedings of the Sixteenth conference on Uncertainty in Artificial Intelligence Proceedings*, pages 46–53, 2000. (Cited on page 80.)

[33] Joseph R. Bourne, Eric R. Pardyjak, and Kam K. Leang. Coordinated Bayesian-Based Bioinspired Plume Source Term Estimation and Source Seeking for Mobile Robots. *IEEE Transactions on Robotics*, 35(4):967–986, 2019. (Cited on page 21.)

[34] Valentino Braitenberg. *Vehicles: Experiments in Synthetic Psychology.* MIT Press, 1986. (Cited on pages 18 and 86.)

[35] David J. Briggs, Susan Collins, Paul Elliott, Paul Fischer, Simon Kingham, Erik Lebret, Karel Pryl, Hans Van Reeuwijk, Kirsty Smallbone, and Andre Van Der Veen. Mapping urban air pollution using GIS: A regression-based approach. *International Journal of Geographical Information Science*, 11(7):699–718, 1997. (Cited on pages 29 and 30.)

[36] Javier Burgués, Victor Hernández, Achim J. Lilienthal, and Santiago Marco. Smelling nano aerial vehicle for gas source localization and mapping. *Sensors*, 19(3), 2019. (Cited on page 19.)

[37] E.J. Candes and Michael B Wakin. An Introduction To Compressive Sampling. *Signal Processing Magazine, IEEE*, 25(2):21–30, 2008. (Cited on page 47.)

[38] Claudio Canuto, M Yousuff Hussaini, Alfio Quarteroni, and Thomas A Zang. *Spectral Methods in Fluid Dynamics*. Springer-Verlag Berlin Heidelberg, 1988. (Cited on page 39.)

[39] José A. Carta, Celia Bueno, and Penélope Ramírez. Statistical modelling of directional wind speeds using mixtures of von Mises distributions: Case study. *Energy Conversion and Management*, 49(5):897–907, 2008. (Cited on page 59.)

[40] Jose A. Carta and Penelope Ramírez. Analysis of two-component mixture Weibull statistics for estimation of wind speed distributions. *Renewable Energy*, 32:518–531, 2007. (Cited on page 59.)

[41] Chih Chung Chang, Jia Lin Wang, Chih Yuan Chang, Mao Chang Liang, and Ming Ren Lin. Development of a multicopter-carried whole air sampling apparatus and its applications in environmental studies. *Chemosphere*, 144:484–492, 2016. (Cited on pages 19 and 22.)

[42] Rick Chartrand and Valentina Staneva. Restricted isometry properties and nonconvex compressive sensing. *Inverse Problems*, 24(3), 2008. (Cited on page 47.)

[43] Jack Chou. *Hazardous Gas Monitors: A Practical Guide to Selection, Operation and Applications*. McGraw-Hill Education, 1999. (Cited on page 14.)

[44] Richard S. Collett and Kehinde Oduyemi. Air quality modelling: A technical review of mathematical approaches. *Meteorological Applications*, 4:235–246, 1997. (Cited on pages 26 and 27.)

[45] Juan D. Colmenares, Omar D. López, and Sergio Preidikman. Computational study of a transverse rotor aircraft in hover using the unsteady vortex lattice method. *Mathematical Problems in Engineering*, 2015, 2015. (Cited on page 19.)

[46] John Crank. *The Mathematics Of Diffusion*. Clarendon Press, Oxford, second edition, 1975. (Cited on page 32.)

[47] Wolfgang Dahmen and Arnold Reusken. *Numerik für Ingenieure und Naturwissenschaftler*. Springer-Verlag Berlin Heidelberg, 2008. (Cited on pages 34, 37, and 38.)

[48] Leonard David. Remote telepresence: A new tool for space exploration? *Aerospace America*, 51(2):38–44, 2013. (Cited on page 2.)

[49] P. de Haan and M. W. Rotach. A puff-particle dispersion model. *International Journal of Environment and Pollution*, 5:350–359, 1995. (Cited on page 27.)

[50] Eddy De Rademaeker, Bruno Fabiano, Simberto Senni Buratti, Pierre Lauret, Frédéric Heymes, Laurent Aprin, Anne Johannet, Gilles Dusserre, Laurent Munier, and Emmanuel Lapébie. Near Field Atmospheric Dispersion Modeling on an Industrial Site Using Neural Networks. *Chemical Engineering Transactions*, 31:151–156, 2013. (Cited on page 31.)

[51] Michael A. Demetriou, Tatiana Egorova, and Nikolaos A. Gatsonis. Estimation of a gaseous release into the atmosphere using a formation of UAVs. *IFAC-PapersOnLine*, 49(18):110–115, 2016. (Cited on pages 7 and 88.)

[52] Michael A Demetriou and Dariusz Ucinski. State Estimation of Spatially Distributed Processes Using Mobile Sensing Agents. In *American Control Conference (ACC)*, pages 1770–1776, 2011. (Cited on page 88.)

[53] David L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006. (Cited on page 47.)

[54] Kok Seng Eu, Kian Meng Yap, and Tiam Hee Tee. An Airflow Analysis Study of Quadrotor Based Flying Sniffer Robot. *Applied Mechanics and Materials*, 627:246–250, 2014. (Cited on page 19.)

[55] Orestis Evangelatos and Jose Rolim. An Airborne Wireless Sensor System for Near-Real Time Air Pollution Monitoring. *Sensors & Transducers*, 189(6):12–21, 2015. (Cited on page 19.)

[56] Jay A. Farrell, Shuo Pang, and Wei Li. Plume Mapping via Hidden Markov Methods. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 33(6):850–863, 2003. (Cited on page 87.)

[57] Thomas Foken. *Micrometeorology*. Springer-Verlag, Berlin Heidelberg, 2008. (Cited on pages 26, 28, 29, and 59.)

[58] Terrence Fong, Jennifer Rochlis Zumbado, Nancy Currie, Andrew Mishkin, and David L. Akin. Space Telerobotics: Unique Challenges to Human-Robot Collaboration in Space. *Reviews of Human Factors and Ergonomics*, 9:6–56, 2013. (Cited on page 2.)

[59] Simon Foucart and Holger Rauhut. *A Mathematical Introduction to Compressive Sensing*. Springer New York Heidelberg Dordrecht London, 2013. (Cited on page 47.)

[60] Brendan Frey and David MacKay. A Revolution: Belief Propagation in Graphs With Cycles. *Advances in Neural Information Processing Systems*, 10:479–485, 1998. (Cited on page 67.)

[61] Brendan J. Frey and Frank R. Kschischang. Probability Propagation and Iterative Decoding. In *Proceedings of the 34th Allerton Conference on Communications, Control and Computing*, 1996. (Cited on page 64.)

[62] Felix Friedl-Vallon, Guido Maucher, Meinhard Seefeldner, Olaf Trieschmann, Anne Kleinert, Anton Lengel, Corneli Keim, Hermann Oelhaf, and Herbert Fischer. Design and characterization of the balloonborne Michelson Interferometer for Passive Atmospheric Sounding

(MIPAS-B2). *Applied Optics*, 43(16):3335–3355, 2004. (Cited on page 20.)

[63] M. B. Frish, M. C. Laderer, R. T. Wainner, A. O. Wright, A. H. Patel, J. Stafford-Evans, J. R. Morency, M. G. Allen, and B. D. Green. The next generation of TDLAS analyzers. *Next-Generation Spectroscopic Technologies SPIE*, 6765:40–49, 2007. (Cited on page 16.)

[64] Bo Gao, Hongbo Li, and Fuchun Sun. 3D moth-inspired chemical plume tracking. In *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 1786–1791, 2015. (Cited on page 21.)

[65] Julian W. Gardner and Philip N. Bartlett. *Electronic Noses Principles and Applications*. Oxford University Press, 1999. (Cited on pages 13 and 111.)

[66] Michael W Gardner and S.R. Dorling. Neural network modelling and prediction of hourly NOx and NO2 concentrations in urban air in London. *Atmospheric Environment*, 33:709–719, 1999. (Cited on pages 30 and 31.)

[67] Ritwik Giri and Bhaskar Rao. Type i and Type II Bayesian Methods for Sparse Signal Recovery Using Scale Mixtures. *IEEE Transactions on Signal Processing*, 64(13):3418–3428, 2016. (Cited on pages 47 and 59.)

[68] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, fourth edition, 2013. (Cited on page 69.)

[69] D. W. Gong, Y. Zhang, and C. L. Qi. Localising odour source using multi-robot and anemotaxis-based particle swarm optimisation. *IET Control Theory & Applications*, 6(11):1661, 2012. (Cited on pages 21, 22, and 86.)

[70] Hadi Hajieghrary, M. Ani Hsieh, and Ira B. Schwartz. Multi-agent search for source localization in a turbulent medium. *Physics Letters, Section A*, 380(20):1698–1705, 2016. (Cited on pages 6, 21, and 87.)

[71] Steve Hankey and Julian D. Marshall. Land Use Regression Models of On-Road Particulate Air Pollution (Particle Number, Black Carbon, PM2.5, Particle Size) Using Mobile Monitoring. *Environmental Science & Technology*, 49(15):9194–9202, 2015. (Cited on pages 22 and 29.)

[72] Thomas Lundgaard Hansen, Bernard Henri Fleury, and Bhaskar D. Rao. Superfast Line Spectral Estimation. *IEEE Transactions on Signal Processing*, 66(10):2511–2526, 2018. (Cited on page 59.)

[73] William A. Harrison, David J. Lary, Brian J. Nathan, and Alec G. Moore. Using remote control aerial vehicles to study variability of airborne particulates. *Air, Soil and Water Research*, 8:43–51, 2015. (Cited on page 18.)

[74] David J. Harvey, Tien Fu Lu, and Michael A. Keller. Comparing insect-inspired chemical plume tracking algorithms using a mobile robot. *IEEE Transactions on Robotics*, 24(2):307–317, 2008. (Cited on page 21.)

[75] Adam T. Hayes, Alcherio Martinoli, and Rodney M. Goodman. Distributed odor source localization. *IEEE Sensors Journal*, 2(3):260–271, 2002. (Cited on pages 20 and 21.)

[76] Xiang He, Joseph R. Bourne, Jake A. Steiner, Cole Mortensen, Kyle C. Hoffman, Christopher J. Dudley, Ben Rogers, Donald M. Cropek, and Kam K. Leang. Autonomous Chemical-Sensing Aerial Robot for Urban/Suburban Environmental Monitoring. *IEEE Systems Journal*, 13(3):3524–3535, 2019. (Cited on page 18.)

[77] Roland Herzog, Georg Stadler, and Gerd Wachsmuth. Directional Sparsity in Optimal Control of Partial Differential Equations. *SIAM Journal on Control and Optimization*, 50(2):943–963, 2012. (Cited on page 47.)

[78] Gerard Hoek, Rob Beelen, Kees de Hoogh, Danielle Vienneau, John Gulliver, Paul Fischer, and David Briggs. A review of land-use regression models to assess spatial variation of outdoor air pollution. *Atmospheric Environment*, 42(33):7561–7578, 2008. (Cited on page 29.)

[79] Khandaker M. A. Hossain. Predictive Ability of Improved Neural Network Models to Simulate Pollutant Dispersion. *International Journal of Atmospheric Sciences*, 2014:1–12, 2014. (Cited on page 31.)

[80] Michael Hutchinson, Cunjia Liu, and Wen Hua Chen. Source term estimation of a hazardous airborne release using an unmanned aerial vehicle. *Journal of Field Robotics*, 36:797–817, 2019. (Cited on page 18.)

[81] Kousuke Ihokura and Joseph Watson. *The stannic oxide gas sensor : principles and applications*. CRC Press, 1994. (Cited on page 111.)

[82] H. Ishida, Y. Kagawa, T. Nakamoto, and T. Moriizumi. Odor-source Localization In Clean Room By Autonomous Mobile Sensing System. *Sensors and Actuators B: Chemical*, 33(1):115–121, 1995. (Cited on pages 21 and 86.)

[83] H. Ishida, K. Suetsugu, T. Nakamoto, and T. Moriizumi. Study of autonomous mobile sensing system for localization of odor source using gas sensors and anemometric sensors. *Sensors and Actuators: A. Physical*, 45:153–157, 1994. (Cited on pages 21 and 87.)

[84] Hiroshi Ishida. Blimp robot for three-dimensional gas distribution mapping in indoor environment. *AIP Conference Proceedings*, 1137:61–64, 2009. (Cited on page 19.)

[85] Hiroshi Ishida, Gouki Nakayama, Takamichi Nakamoto, and Toyosaka Moriizumi. Controlling a gas/odor plume-tracking robot based on transient responses of gas sensors. *IEEE Sensors Journal*, 2(3):1665–1670, 2002. (Cited on page 21.)

[86] Hiroshi Ishida, Yuta Wada, and Haruka Matsukura. Chemical sensing in robotic applications: A review. *IEEE Sensors Journal*, 12(11):3163–3173, 2012. (Cited on pages 17, 20, 21, 22, and 85.)

[87] Gideon Kowadlo and R. Andrew Russell. Robot odor localization: A taxonomy and survey. *International Journal of Robotics Research*, 27(8):869–894, 2008. (Cited on pages 85 and 86.)

[88] Frank R. Kschischang, Brendan J. Frey, and Hans Andrea Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47(2):498–519, 2001. (Cited on pages 7, 64, 66, 67, and 69.)

[89] Carlos. S. Kubrusly and Helios Malebranche. Sensors and controllers location in distributed systems-A survey. *Automatica*, 21(2):117–128, 1985. (Cited on page 88.)

[90] Solomon Kullback and Richard A. Leibler. On Information and Sufficiency. *Annals of Mathematical Statistics*, 22(1):79–86, 1951. (Cited on page 70.)

[91] Karl Kunisch, Konstantin Pieper, and Boris Vexler. Measure Valued Directional Sparsity for Parabolic Optimal Control Problems. *SIAM Journal on Control and Optimization*, 52(5):3078–3108, jan 2014. (Cited on page 47.)

[92] Yuki Kuroki, George S Young, and Sue Ellen. Expert Systems with Applications UAV navigation by an expert system for contaminant mapping with a genetic algorithm. *Expert Systems With Applications*, 37(6):4687–4697, 2010. (Cited on page 27.)

[93] Maximilian Lackner. Tunable diode laser absorption spectroscopy (TDLAS) in the process industries - A review. *Reviews in Chemical Engineering*, 23(2):65–147, 2007. (Cited on page 16.)

[94] Ádám Leelossy, Ferenc Molnár Jr., Ferenc Izsák, Ágnes Havasi, István Lagzi, and Róbert Mészáros. Dispersion modeling of air pollutants in the atmosphere: a review. *Central European Journal of Geosciences*, 6(3):257–278, 2014. (Cited on pages 26, 27, and 28.)

[95] Shuai Li, Yi Guo, and Brian Bingham. Multi-robot cooperative control for monitoring and tracking dynamic plumes. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 67–73, 2014. (Cited on pages 22 and 88.)

[96] Achim Lilienthal and Tom Duckett. A stereo electronic nose for a mobile inspection robot. In *IEEE International Workshop on Robotic Sensing (ROSE)*, page 6, 2003. (Cited on pages 13 and 111.)

[97] Achim Lilienthal and Tom Duckett. Gas Source Localisation by Constructing Concentration Gridmaps with a Mobile Robot. In *European Conference on Mobile Robots (ECMR)*, pages 1–6, 2003. (Cited on page 30.)

[98] Achim Lilienthal and Tom Duckett. Building gas concentration gridmaps with a mobile robot. *Robotics and Autonomous Systems*, 48:3–16, 2004. (Cited on pages 6, 22, 30, and 85.)

[99] Achim Lilienthal and Tom Duckett. Experimental analysis of gas-sensitive Braitenberg vehicles. *Advanced Robotics*, 18(8):817–834, 2004. (Cited on pages 18, 21, and 86.)

[100] Achim Lilienthal, Denis Reimann, and Andreas Zell. Gas Source Tracing with a Mobile Robot Using an Adapted Moth Strategy. In *Autonomous Mobile Systems*, pages 150–160, 2003. (Cited on pages 21 and 87.)

[101] Achim Lilienthal, Holger Ulmer, Holger Fr, Andreas St, Felix Werner, and Andreas Zell. Gas Source Declaration with a Mobile Robot. In *IEEE International Conference on Robotic and Automation, (ICRA)*, pages 1430–1435, 2004. (Cited on page 15.)

[102] Achim J. Lilienthal, Tom Duckett, Hiroshi Ishida, and Felix Werner. Indicators of gas source proximity using metal oxide sensors in a turbulent environment. In *IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, pages 733–738, 2006. (Cited on page 111.)

[103] Achim J. Lilienthal, Amy Loutfi, and Tom Duckett. Airborne chemical sensing with mobile robots. *Sensors*, 6(11):1616–1678, 2006. (Cited on pages 15 and 85.)

[104] Achim J Lilienthal, Matteo Reggente, Marco Trincavelli, Jose Luis Blanco, and Javier Gonzalez. A statistical approach to gas distribution modelling with mobile robots-the kernel dm+ v algorithm. In *IEEE International Conference on Intelligent Robots and Systems (IROS)*, pages 570–576, 2009. (Cited on pages 30 and 60.)

[105] D. K. Lilly. The representation of small-scale turbulence in numerical simulation experiments. In *Proceedings of IBM Scientific Computing Symposium on Environmental Sciences*, pages 195–210, 1967. (Cited on page 29.)

[106] Thomas Lochmatter and Alcherio Martinoli. Tracking Odor Plumes in a Laminar Wind Field with Bio-inspired Algorithms. *Experimental Robotics, Springer Tracts in Advanced Robotics*, 54:473–482, 2009. (Cited on pages 21 and 87.)

[107] Thomas Lochmatter, Xavier Raemy, Loïc Matthey, Saurabh Indra, and Alcherio Martinoli. A comparison of casting and spiraling algorithms for odor source localization in laminar flow. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1138–1143, 2008. (Cited on page 21.)

[108] Hans-Andrea Loeliger. An Introduction to Factor Graphs. *IEEE Signal Processing Magazine*, 21:28 – 41, 2004. (Cited on pages 64, 66, and 67.)

[109] Anders Logg, Kent Mardal, and Garth Wells, editors. *Automated Solution of Differential Equations by the Finite Element Method*. Springer, 2012. (Cited on page 98.)

[110] Bing Luo, Qing Hao Meng, Jia Ying Wang, and Shu Gen Ma. Simulate the aerodynamic olfactory effects of gas-sensitive UAVs: A numerical model and its parallel implementation. *Advances in Engineering Software*, 102:123–133, 2016. (Cited on pages 18 and 19.)

[111] Bing Luo, Qing Hao Meng, Jia Ying Wang, Biao Sun, and Ying Wang. Three-dimensional gas distribution mapping with a micro-drone. In *Chinese Control Conference (CCC)*, pages 6011–6015. Technical Committee on Control Theory, Chinese Association of Automation, 2015. (Cited on pages 18 and 22.)

[112] Ali Marjovi and Lino Marques. Optimal spatial formation of swarm robotic gas sensors in odor plume finding. *Autonomous Robots*, 35:93–109, 2013. (Cited on pages 22 and 88.)

[113] Ali Marjovi and Lino Marques. Multi-robot odor distribution mapping in realistic time-variant conditions. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3720–3727, 2014. (Cited on pages 22, 86, and 87.)

[114] Maria Markiewicz. A review of mathematical models for the atmospheric dispersion of heavy gases. Part I. A classification of models. *Ecological Chemistry and Engineering S*, 19:297–314, 2012. (Cited on page 28.)

[115] Lino Marques and A.T. De Almeida. Electronic nose-based odour source localization. *6th International Workshop on Advanced Motion Control Proceedings*, pages 36–40, 2000. (Cited on pages 13, 15, and 86.)

[116] Marta Martinez-Camara, Ivan Dokmanic, Juri Ranieri, Robin Scheibler, Martin Vetterli, and Andreas Stohl. The Fukushima Inverse Problem. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4330–4334, 2013. (Cited on page 47.)

[117] Andrew J S McGonigle, A. Aiuppa, G. Giudice, G. Tamburello, A. J. Hodson, and S. Gurrieri. Unmanned aerial vehicle measurements of volcanic carbon dioxide fluxes. *Geophysical Research Letters*, 35(6):3–6, 2008. (Cited on pages 3, 14, and 19.)

[118] Qing Hao Meng, Wei Xing Yang, Yang Wang, and Ming Zeng. Collective odor source estimation and search in time-variant airflow environments using mobile robots. *Sensors*, 11:10415–10443, 2011. (Cited on pages 15, 18, 21, 22, and 87.)

[119] Kyle P. Messier, Sarah E. Chambliss, Shahzad Gani, Ramon Alvarez, Michael Brauer, Jonathan J. Choi, Steven P. Hamburg, Jules Kerckhoffs, Brian Lafranchi, Melissa M. Lunden, Julian D. Marshall, Christopher J. Portier, Ananya Roy, Adam A. Szpiro, Roel C.H. Vermeulen, and Joshua S. Apte. Mapping Air Pollution with Google Street View Cars: Efficient Approaches with Mobile Monitoring and Land Use Regression. *Environmental Science & Technology*, 52:12563–12572, 2018. (Cited on pages 29 and 30.)

[120] Florian Meyer, Erwin Riegler, Ondrej Hlinka, and Franz Hlawatsch. Simultaneous Distributed Sensor Self-Localization and Target Tracking Using Belief Propagation and Likelihood Consensus. In *Conference Record of the Forty Sixth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, pages 1212–1216, 2012. (Cited on page 73.)

[121] Milton R. Beychok. *Fundamentals of Stack Gas Dispersion*. Milton R. Beychok, 1994. (Cited on page 26.)

[122] Javier Monroy, Victor Hernandez-Bennetts, Han Fan, Achim Lilienthal, and Javier Gonzalez-Jimenez. GADEN: A 3D gas dispersion simulator for mobile robot olfaction in realistic environments. *Sensors*, 17(7):1479, 2017. (Cited on pages 28 and 59.)

[123] Javier Monroy, Jose Raul Ruiz-Sarmiento, Francisco Angel Moreno, Francisco Melendez-Fernandez, Cipriano Galindo, and Javier Gonzalez-Jimenez. A semantic-based gas source localization with a mobile robot combining vision and chemical sensing. *Sensors*, 18(12):4174, 2018. (Cited on page 20.)

[124] Javier G Monroy, Jose-Luis Blanco, and Javier Gonzalez-Jimenez. Time-Variant Gas Distribution Mapping with Obstacle Information. *Autonomous Robots*, 40:1–16, 2016. (Cited on pages 16 and 22.)

[125] Javier G. Monroy and Javier Gonzalez-Jimenez. Gas classification in motion: An experimental analysis. *Sensors and Actuators, B: Chemical*, 240:1205–1215, 2017. (Cited on page 18.)

[126] Eduardo Martin Moraud and Dominique Martinez. Effectiveness and robustness of robot infotaxis for searching in dilute conditions. *Frontiers in Neurorobotics*, 4:1–8, 2010. (Cited on pages 6, 21, and 87.)

[127] Kevin P Murphy, Yair Weiss, and Michael I Jordan. Loopy belief propagation for approximate inference: An empirical study. *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, 9:467–475, 1999. (Cited on page 67.)

[128] Brian J. Nathan, Levi M. Golston, Anthony S. O'Brien, Kevin Ross, William A. Harrison, Lei Tao, David J. Lary, Derek R. Johnson, April N. Covington, Nigel N. Clark, and Mark A. Zondlo. Near-Field Characterization of Methane Emission Variability from a Compressor Station Using a Model Aircraft. *Environmental Science & Technology*, 49(13):7896–7903, 2015. (Cited on page 18.)

[129] Patrick P Neumann, Sahar Asadi, Achim J Lilienthal, Matthias Bartholmai, and Jochen H Schiller. Micro-Drone for Wind Vector Estimation and Gas Distribution Mapping. *IEEE Robotics & Automation Magazine*, 19:50–61, 2012. (Cited on page 18.)

[130] Patrick P. Neumann, Victor Hernandez Bennetts, Achim J. Lilienthal, Matthias Bartholmai, and Jochen H. Schiller. Gas source localization with a micro-drone using bio-inspired and particle filter-based algorithms. *Advanced Robotics*, 27(9):725–738, 2013. (Cited on pages 18, 19, 21, 86, and 87.)

[131] Patrick P. Neumann, Harald Kohlhoff, Dino Huellmann, Daniel Krentel, Martin Kluge, Marcin Dzierlinski, Achim J. Lilienthal, and Matthias Bartholmai. Aerial-based gas tomography-from single beams to complex gas distributions. *European Journal of Remote Sensing*, 52:2–16, 2019. (Cited on pages 18 and 23.)

[132] Patrick P. Neumann, Harald Kohlhoff, Dino Hullmann, Achim J. Lilienthal, and Martin Kluge. Bringing Mobile Robot Olfaction to the next dimension - UAV-based remote sensing of gas clouds and source localization. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3910–3916, 2017. (Cited on page 16.)

[133] Christopher Nicol, Alex Ellery, Brian Lynch, Ed Cloutis, Guido De Croon, and Alex Ellery. Martian methane plume models for defining Mars rover methane source search strategies. *International Journal of Astrobiology*, 17:228–238, 2018. (Cited on pages 21 and 86.)

[134] Trevor Park and George Casella. The Bayesian Lasso. *Journal of the American Statistical Association*, 103(482):681–686, 2008. (Cited on page 47.)

[135] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann Publishers, San Francisco, 1988. (Cited on pages 66 and 67.)

[136] Krishna Persaud and George Dodd. Analysis of discrimination mechanisms in the mammalian olfactory system using a model nose. *Nature*, 299:352–355, 1982. (Cited on page 13.)

[137] Ignacio Pisso, Espen Sollum, Henrik Grythe, Nina I. Kristiansen, Massimo Cassiani, Sabine Eckhardt, Delia Arnold, Don Morton, Rona L. Thompson, Christine D. Groot Zwaaftink, Nikolaos Evangeliou, Harald Sodemann, Leopold Haimberger, Stephan Henne, Dominik Brunner, John F. Burkhart, Anne Fouilloux, Jerome Brioude, Anne Philipp, Petra Seibert, and Andreas Stohl. The Lagrangian particle dispersion model FLEXPART version 10.4. *Geoscientific Model Development*, 12:4955–4997, 2019. (Cited on page 28.)

[138] Domagoj Podnar, Darko Koracin, and Anna Panorska. Application of artificial neural networks to modeling the transport and dispersion of tracers in complex terrain. *Atmospheric Environment*, 36:561–570, 2002. (Cited on page 30.)

[139] Friedrich Pukelsheim. *Optimal Design of Experiments*. John Wiley & Sons, Inc., 1993. (Cited on page 88.)

[140] Pawel Pyk, Sergi Bermúdez i Badia, Ulysses Bernardet, Philipp Knüsel, Mikael Carlsson, Jing Gu, Eric Chanie, Bill S. Hansson, Tim C. Pearce, and Paul F.M.J. Verschure. An artificial moth: Chemical source localization using a robot based neuronal model of moth optomotor anemotactic search. *Autonomous Robots*, 20:197–213, 2006. (Cited on page 19.)

[141] Sihang Qiu, Bin Chen, Rongxiao Wang, Zhengqiu Zhu, Yuan Wang, and Xiaogang Qiu. Atmospheric dispersion prediction and source estimation of hazardous gas using artificial neural network, particle swarm

optimization and expectation maximization. *Atmospheric Environment*, 178:158–163, 2018. (Cited on page 30.)

[142] RAE Systems Inc. *The PID Handbook - Theory and Applications of Direct-Reading Photoionization Detectors*. 2013. (Cited on page 15.)

[143] Matteo Reggente. *Statistical Gas Distribution Modelling for Mobile Robot Applications*. PhD thesis, Örebro University, 2014. (Cited on page 26.)

[144] Matteo Reggente and Achim J. Lilienthal. Using local wind information for gas distribution mapping in outdoor environments with a mobile robot. In *Proceedings of IEEE Sensors*, pages 1715–1720. IEEE, 2009. (Cited on pages 18, 20, and 30.)

[145] Matteo Reggente and Achim J. Lilienthal. The 3D-kernel DM+V/W algorithm: Using wind information in three dimensional gas distribution modelling with a mobile robot. In *Proceedings of IEEE Sensors*, pages 999–1004. IEEE, 2010. (Cited on page 17.)

[146] Erwin Riegler, Gunvor E. Kirkelund, Carles N. Manchon, Mihai-Alin Badiu, and Bernard H. Fleury. Merging belief propagation and the mean field approximation: A free energy approach. *IEEE TRANSACTIONS ON INFORMATION THEORY*, 59:588–602, 2013. (Cited on page 66.)

[147] O. F. T. Roberts. The Theoretical Scattering of Smoke in a Turbulent Atmosphere. *Proceedings of the Royal Society of London. Series A*, 104(728):640–654, 1923. (Cited on page 27.)

[148] Robotnik. SUMMIT- XL Data Sheet. Technical report, Robotnik, 2019. (Cited on page 115.)

[149] Juan Duque Rodríguez, David Gómez-Ullate, and Carlos Mejía-Monasterio. On the performance of blind-infotaxis under inaccurate modeling of the environment. *European Physical Journal: Special Topics*, 226:2407–2420, 2017. (Cited on page 87.)

[150] Christoph Rösmann, Wendelin Feiten, Thomas Wösch, Frank Hoffmann, and Torsten Bertram. Trajectory modification considering dynamic constraints of autonomous robots. In *7th German Conference on Robotics*, pages 74–79, 2012. (Cited on page 116.)

[151] Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. A Metric for Distributions with Applications to Image Databases. In *IEEE International Conference on Computer Vision*, pages 59–66, 1998. (Cited on pages 98 and 149.)

[152] Julian Ruddick, Ali Marjovi, Faezeh Rahbar, and Alcherio Martinoli. Design and Performance Evaluation of an Infotaxis-Based Three-Dimensional Algorithm for Odor Source Localization. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1413–1420, 2019. (Cited on pages 6, 7, 21, and 87.)

[153] R. Andrew Russell, Alireza Bab-Hadiashar, Rod L. Shepherd, and Gordon G. Wallace. A comparison of reactive robot chemotaxis algorithms. *Robotics and Autonomous Systems*, 45:83–97, 2003. (Cited on pages 86 and 87.)

[154] R Andrew Russell, David Thiel, Reimundo Deveza, and Alan Mackay-Sim. A robotic system to locate hazardous chemical leaks. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 556–561, 1995. (Cited on page 17.)

[155] Michael Schmuker, Viktor Bahr, and Ramón Huerta. Exploiting plume structure to decode gas source distance using metal-oxide gas sensors. *Sensors and Actuators, B: Chemical*, 235:636–646, 2016. (Cited on page 111.)

[156] Frank E. Schneider and Dennis Wildermuth. Using robots for firefighters and first responders: Scenario specification and exemplary system description. In *18th International Carpathian Control Conference (ICCC)*, pages 216–221, 2017. (Cited on page 2.)

[157] Tetsuro Seiyama, Kiyoshi Fujiishi, Masanoai Nagatani, and Akio Kato. A New Detector for Gaseous Components Using Zinc Oxide Thin Films. *Analytical Chemistry*, 34:1502–1503, 1962. (Cited on page 15.)

[158] Dmitriy Shutin, Thomas Buchgraber, Sanjeev R. Kulkarni, and H. Vincent Poor. Fast variational sparse bayesian learning with automatic relevance determination for superimposed signals. *IEEE Transactions on Signal Processing*, 59(12):6257–6261, 2011. (Cited on page 59.)

[159] Dmitriy Shutin, Sanjeev R. Kulkarni, and H. Vincent Poor. Stationary point variational Bayesian attribute-distributed sparse learning with l 1 sparsity constraints. In *IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pages 277–280, 2011. (Cited on page 70.)

[160] Asif A Siddiqi. *Beyond Earth: a chronicle of deep space exploration, 1958-2016*. Office of Communications, NASA History Division, 2018. (Cited on page 1.)

[161] Joseph Smagorinsky. Circulation Experiments With the Primitive Equations: I. the Basic Experiment. *Monthly Weather Review*, 91(3), 1963. (Cited on page 29.)

[162] Jorge M Soares, A Pedro Aguiar, and Alcherio Martinoli. An Algorithm for Formation-Based Chemical Plume Tracing Using Robotic Marine Vehicles. In *OCEANS 2016 MTS/IEEE Monterey*, pages 1–8, 2016. (Cited on pages 22 and 86.)

[163] Jorge M. Soares, A. Pedro Aguiar, Antonio M. Pascoal, and Alcherio Martinoli. A distributed formation-based odor source localization algorithm - Design, implementation, and wind tunnel evaluation. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1830–1836, 2015. (Cited on pages 21 and 88.)

[164] John Stix, J. Maarten de Moor, Julian Rüdiger, Alfredo Alan, Ernesto Corrales, Fiona D'Arcy, Jorge Andres Diaz, and Marcello Liotta. Using Drones and Miniaturized Instrumentation to Study Degassing at Turrialba and Masaya Volcanoes, Central America. *Journal of Geophysical Research: Solid Earth*, 123:6501–6520, 2018. (Cited on page 18.)

[165] John M Stockie. The Mathematics of Atmospheric Dispersion Modeling. *SIAM Review*, 53(2):349–372, 2011. (Cited on page 26.)

[166] John C. Strikwerda. *Finite Difference Schemes and Partial Differential Equations*. Society for Industrial and Applied Mathematics, Philadelphia, second edition, 2004. (Cited on pages 33, 34, and 35.)

[167] O. G. Sutton. A Theory of Eddy Diffusion in the Atmosphere. *Proceedings of the Royal Society of London. Series A*, 135(826):143–165, 1932. (Cited on page 27.)

[168] Yoshinori Takei, Yukitaka Kanazawa, Kazuki Hirasawa, and Hidehito Nanto. Development of 3D gas source localization using multi-copter with gas sensor array. In *18th International Symposium on Olfaction and Electronic Nose (ISOEN)*, pages 2–5. IEEE, 2019. (Cited on page 18.)

[169] Howard Talkington. Manned and remotely operated submersible systems: A comparison. *Elsevier Oceanography Series*, 17(C):77–95, 1977. (Cited on page 1.)

[170] R. Michael Tanner. A Recursive Approach to Low Complexity Codes. *IEEE Transactions on Information Theory*, 27(5):533–547, 1981. (Cited on page 64.)

[171] R Thompson. Numeric calculation of turbulent diffusion. *Quarterly Journal of the Royal Meteorological Society*, 97(411):93–98, 1971. (Cited on page 28.)

[172] Nils Thuerey and Tobias Pfaff. MantaFlow. *http://mantaflow.com/ (Date Accessed 2019-11-2)*, 2016. (Cited on page 99.)

[173] Robert Tibshirani. Regression Shrinkage and Selection Via the Lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996. (Cited on page 47.)

[174] Michael E. Tipping. Sparse Bayesian Learning and the Relevance Vector Machine. *Journal of Machine Learning Research*, 1:211–244, 2001. (Cited on pages 58 and 82.)

[175] Michael E. Tipping and Anita C. Faul. Fast Marginal Likelihood Maximization for Sparse Bayesian Models. In *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*, pages 1–5, 2003. (Cited on page 59.)

[176] Christophe Tricaud, Maciej Patan, Dariusz Uciński, and Yang Quan Chen. D-optimal trajectory design of heterogeneous mobile sensors for parameter estimation of distributed systems. In *Proceedings of the American Control Conference*, pages 663–668, 2008. (Cited on page 88.)

[177] Marco Trincavelli, Victor Hernandez Bennetts, and Achim J. Lilienthal. A least squares approach for learning gas distribution maps from a set of integral gas concentration measurements obtained with a TDLAS sensor. In *IEEE Sensors*, pages 1–4. IEEE, 2012. (Cited on page 23.)

[178] Dariusz Uciński. *Measurement optimization for parameter estimation in distributed systems*. Technical University Press, Zielona Gora, 1999. (Cited on page 88.)

[179] Dariusz Ucinski and Przemysław Baranowski. A Parallel Algorithm for Optimum Monitoring Network Design in Parameter Estimation of Distributed Systems. In *European Control Conference (ECC)*, pages 1609–1614, 2013. (Cited on page 88.)

[180] Massimo Vergassola, Emmanuel Villermaux, and Boris I. Shraiman. 'Infotaxis' as a strategy for searching without gradients. *Nature*, 445(7126):406–409, 2007. (Cited on pages 6 and 87.)

[181] Henk Kaarle Versteeg and Weeratunge Malalasekera. *An Introduction to Computational Fluid Dynamics*. Pearson Education Limited, second edition, 2007. (Cited on pages 27, 28, and 29.)

[182] Tommaso Villa, Felipe Gonzalez, Branka Miljievic, Zoran Ristovski, and Lidia Morawska. An Overview of Small Unmanned Aerial Vehicles for Air Quality Measurements: Present Applications and Future Prospectives. *Sensors*, 16(7):1072, 2016. (Cited on pages 19 and 22.)

[183] Tommaso Francesco Villa, Farhad Salimi, Kye Morton, Lidia Morawska, and Felipe Gonzalez. Development and validation of a UAV based system for air pollution measurements. *Sensors*, 16(12):2202, 2016. (Cited on page 14.)

[184] Alberto Viseras and Ricardo Garcia. DeepIG: Multi-robot information gathering with deep reinforcement learning. *IEEE Robotics and Automation Letters*, 4(3):3059–3066, 2019. (Cited on page 144.)

[185] Mikel Vuka, Erik Schaffernicht, Michael Schmuker, Victor Hernandez Bennetts, Francesco Amigoni, and Achim J. Lilienthal. Exploration and localization of a gas source with MOX gas sensors on a mobile robot-A Gaussian regression bout amplitude approach. In *ISOCS/IEEE International Symposium on Olfaction and Electronic Nose (ISOEN9*, pages 3–5, 2017. (Cited on page 15.)

[186] T. Watai, T. Machida, N. Ishizaki, G. Inoue, T. Watai, T. Machida, N. Ishizaki, and G. Inoue. A Lightweight Observation System for Atmospheric Carbon Dioxide Concentration Using a Small Unmanned Aerial Vehicle. *Journal of Atmospheric and Oceanic Technology*, 23(5):700–710, 2006. (Cited on page 18.)

[187] Joseph Watson. The tin oxide gas sensor and its applications. *Sensors and Actuators*, 5(1):29–42, 1984. (Cited on page 15.)

[188] Thomas Wiedemann, Achim J. Lilienthal, and Dmitriy Shutin. Analysis of model mismatch effects for a model-based gas source localization strategy incorporating advection knowledge. *Sensors*, 19(3), 2019. (Cited on pages 10, 79, 97, and 100.)

[189] Thomas Wiedemann, Christoph Manss, and Dmitriy Shutin. Multiagent exploration of spatial dynamical processes under sparsity constraints. *Autonomous Agents and Multi-Agent Systems*, 32(1):134–162, 2018. (Cited on pages 9 and 66.)

[190] Thomas Wiedemann, Christoph Manss, Dmitriy Shutin, Achim J. Lilienthal, Valentina Karolj, and Alberto Viseras. Probabilistic modeling of gas diffusion with partial differential equations for multi-robot exploration and gas source localization. In *European Conference on Mobile Robots (ECMR)*, pages 1–7, 2017. (Cited on pages 10, 107, and 117.)

[191] Thomas Wiedemann, Dmitriy Shutin, Victor Hernandez, Erik Schaffernicht, and Achim J. Lilienthal. Bayesian gas source localization and exploration with a multi-robot system using partial differential equation based modeling. In *ISOCS/IEEE International Symposium on Olfaction and Electronic Nose (ISOEN)*, pages 8–10, 2017. (Cited on page 10.)

[192] 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. *Journal of Robotics*

and Autonomous Systems, 118:66–79, 2019. (Cited on pages 10 and 117.)

[193] John Winn and Christopher M Bishop. Variational Message Passing. *Journal of Machine Learning Research*, 6:661–694, 2005. (Cited on pages 69 and 70.)

[194] David Wipf and Srikantan Nagarajan. A New View of Automatic Relevance Determination. *Advances in Neural Information Processing Systems 20*, pages 1625–1632, 2008. (Cited on pages 58 and 59.)

[195] David P. Wipf and Bhaskar D. Rao. Sparse Bayesian learning for basis selection. *IEEE Transactions on Signal Processing*, 52(8):2153–2164, 2004. (Cited on page 58.)

[196] Xin xing Chen and Jian Huang. Odor source localization algorithms on mobile robots: A review and future outlook. *Robotics and Autonomous Systems*, 112:123–136, 2019. (Cited on pages 85 and 86.)

[197] Feng Bo Yang, Xin Yu Xue, Ling Zhang, and Zhu Sun. Numerical simulation and experimental verification on downwash air flow of six-rotor agricultural unmanned aerial vehicle in hover. *International Journal of Agricultural and Biological Engineering*, 10(4):41–53, 2017. (Cited on page 19.)

[198] Byungtae Yoo and Sang D. Choi. Emergency evacuation plan for hazardous chemicals leakage accidents using GIS-based risk analysis techniques in South Korea. *International Journal of Environmental Research and Public Health*, 16(11), 2019. (Cited on page 26.)

[199] Paolo Zannetti. *Air Pollution Modeling: Theories, Computational Methods and Available Software*. Springer, 1990. (Cited on page 27.)

[200] Paolo Zannetti. Numerical simulation modeling of air pollution: an overview. *WIT Transactions on Ecology and the Environment*, 1, 1993. (Cited on page 27.)

[201] Siqi Zhang and Demin Xu. A survey of biologically inspired chemical plume tracking strategies for single robot in 2-D turbulence dominated flow environments. In *IEEE/SICE International Symposium on System Integration (SII)*, pages 348–353, 2011. (Cited on page 85.)

[202] Yang Zhiyin. Large-eddy simulation: Past, present and the future. *Chinese Journal of Aeronautics*, 28(1):11–24, 2015. (Cited on pages 28 and 29.)

[203] Olek C. Zienkiewicz, Robert L. Taylor, and J. Z. Zhu. *The Finite Element Method: Its Basis and Fundamentals*. Elsevier Butterworth-Heinemann, sixth edition, 2013. (Cited on pages 34, 37, and 38.)

Publications *in the series*
Örebro Studies in Technology

1. Bergsten, Pontus (2001) *Observers and Controllers for Takagi – Sugeno Fuzzy Systems*. Doctoral Dissertation.
2. Iliev, Boyko (2002) *Minimum-time Sliding Mode Control of Robot Manipulators*. Licentiate Thesis.
3. Spännar, Jan (2002) *Grey box modelling for temperature estimation*. Licentiate Thesis.
4. Persson, Martin (2002) *A simulation environment for visual servoing*. Licentiate Thesis.
5. Boustedt, Katarina (2002) *Flip Chip for High Volume and Low Cost – Materials and Production Technology*. Licentiate Thesis.
6. Biel, Lena (2002) *Modeling of Perceptual Systems – A Sensor Fusion Model with Active Perception*. Licentiate Thesis.
7. Otterskog, Magnus (2002) *Produktionstest av mobiltelefonantenner i mod-växlande kammare*. Licentiate Thesis.
8. Tolt, Gustav (2003) *Fuzzy-Similarity-Based Low-level Image Processing*. Licentiate Thesis.
9. Loutfi, Amy (2003) *Communicating Perceptions: Grounding Symbols to Artificial Olfactory Signals*. Licentiate Thesis.
10. Iliev, Boyko (2004) *Minimum-time Sliding Mode Control of Robot Manipulators*. Doctoral Dissertation.
11. Pettersson, Ola (2004) *Model-Free Execution Monitoring in Behavior-Based Mobile Robotics*. Doctoral Dissertation.
12. Överstam, Henrik (2004) *The Interdependence of Plastic Behaviour and Final Properties of Steel Wire, Analysed by the Finite Element Metod*. Doctoral Dissertation.
13. Jennergren, Lars (2004) *Flexible Assembly of Ready-to-eat Meals*. Licentiate Thesis.
14. Jun, Li (2004) *Towards Online Learning of Reactive Behaviors in Mobile Robotics*. Licentiate Thesis.
15. Lindquist, Malin (2004) *Electronic Tongue for Water Quality Assessment*. Licentiate Thesis.
16. Wasik, Zbigniew (2005) *A Behavior-Based Control System for Mobile Manipulation*. Doctoral Dissertation.

17. Berntsson, Tomas (2005) *Replacement of Lead Baths with Environment Friendly Alternative Heat Treatment Processes in Steel Wire Production.* Licentiate Thesis.

18. Tolt, Gustav (2005) *Fuzzy Similarity-based Image Processing.* Doctoral Dissertation.

19. Munkevik, Per (2005) *"Artificial sensory evaluation – appearance-based analysis of ready meals"*. Licentiate Thesis.

20. Buschka, Pär (2005) *An Investigation of Hybrid Maps for Mobile Robots.* Doctoral Dissertation.

21. Loutfi, Amy (2006) *Odour Recognition using Electronic Noses in Robotic and Intelligent Systems.* Doctoral Dissertation.

22. Gillström, Peter (2006) *Alternatives to Pickling; Preparation of Carbon and Low Alloyed Steel Wire Rod.* Doctoral Dissertation.

23. Li, Jun (2006) *Learning Reactive Behaviors with Constructive Neural Networks in Mobile Robotics.* Doctoral Dissertation.

24. Otterskog, Magnus (2006) *Propagation Environment Modeling Using Scattered Field Chamber.* Doctoral Dissertation.

25. Lindquist, Malin (2007) *Electronic Tongue for Water Quality Assessment.* Doctoral Dissertation.

26. Cielniak, Grzegorz (2007) *People Tracking by Mobile Robots using Thermal and Colour Vision.* Doctoral Dissertation.

27. Boustedt, Katarina (2007) *Flip Chip for High Frequency Applications – Materials Aspects.* Doctoral Dissertation.

28. Soron, Mikael (2007) *Robot System for Flexible 3D Friction Stir Welding.* Doctoral Dissertation.

29. Larsson, Sören (2008) *An industrial robot as carrier of a laser profile scanner: Motion control, data capturing and path planning.* Doctoral Dissertation.

30. Persson, Martin (2008) *Semantic Mapping Using Virtual Sensors and Fusion of Aerial Images with Sensor Data from a Ground Vehicle.* Doctoral Dissertation.

31. Andreasson, Henrik (2008) *Local Visual Feature based Localisation and Mapping by Mobile Robots.* Doctoral Dissertation.

32. Bouguerra, Abdelbaki (2008) *Robust Execution of Robot Task-Plans: A Knowledge-based Approach.* Doctoral Dissertation.

33. Lundh, Robert (2009) *Robots that Help Each Other: Self-Configuration of Distributed Robot Systems.* Doctoral Dissertation.

34. Skoglund, Alexander (2009) *Programming by Demonstration of Robot Manipulators*. Doctoral Dissertation.

35. Ranjbar, Parivash (2009) *Sensing the Environment: Development of Monitoring Aids for Persons with Profound Deafness or Deafblindness*. Doctoral Dissertation.

36. Magnusson, Martin (2009) *The Three-Dimensional Normal-Distributions Transform – an Efficient Representation for Registration, Surface Analysis, and Loop Detection.* Doctoral Dissertation.

37. Rahayem, Mohamed (2010) *Segmentation and fitting for Geometric Reverse Engineering. Processing data captured by a laser profile scanner mounted on an industrial robot.* Doctoral Dissertation.

38. Karlsson, Alexander (2010) *Evaluating Credal Set Theory as a Belief Framework in High-Level Information Fusion for Automated Decision-Making*. Doctoral Dissertation.

39. LeBlanc, Kevin (2010) *Cooperative Anchoring – Sharing Information About Objects in Multi-Robot Systems*. Doctoral Dissertation.

40. Johansson, Fredrik (2010) *Evaluating the Performance of TEWA Systems*. Doctoral Dissertation.

41. Trincavelli, Marco (2010) *Gas Discrimination for Mobile Robots.* Doctoral Dissertation.

42. Cirillo, Marcello (2010) *Planning in Inhabited Environments: Human-Aware Task Planning and Activity Recognition.* Doctoral Dissertation.

43. Nilsson, Maria (2010) *Capturing Semi-Automated Decision Making: The Methodology of CASADEMA*. Doctoral Dissertation.

44. Dahlbom, Anders (2011) *Petri nets for Situation Recognition*. Doctoral Dissertation.

45. Ahmed, Muhammad Rehan (2011) *Compliance Control of Robot Manipulator for Safe Physical Human Robot Interaction.* Doctoral Dissertation.

46. Riveiro, Maria (2011) *Visual Analytics for Maritime Anomaly Detection*. Doctoral Dissertation.

47. Rashid, Md. Jayedur (2011) *Extending a Networked Robot System to Include Humans, Tiny Devices, and Everyday Objects*. Doctoral Dissertation.

48. Zain-ul-Abdin (2011) *Programming of Coarse-Grained Reconfigurable Architectures*. Doctoral Dissertation.

49. Wang, Yan (2011) *A Domain-Specific Language for Protocol Stack Implementation in Embedded Systems*. Doctoral Dissertation.

50. Brax, Christoffer (2011) *Anomaly Detection in the Surveillance Domain*. Doctoral Dissertation.

51. Larsson, Johan (2011) *Unmanned Operation of Load-Haul-Dump Vehicles in Mining Environments*. Doctoral Dissertation.

52. Lidström, Kristoffer (2012) *Situation-Aware Vehicles: Supporting the Next Generation of Cooperative Traffic Systems*. Doctoral Dissertation.

53. Johansson, Daniel (2012) *Convergence in Mixed Reality-Virtuality Environments. Facilitating Natural User Behavior*. Doctoral Dissertation.

54. Stoyanov, Todor Dimitrov (2012) *Reliable Autonomous Navigation in Semi-Structured Environments using the Three-Dimensional Normal Distributions Transform (3D-NDT)*. Doctoral Dissertation.

55. Daoutis, Marios (2013) *Knowledge Based Perceptual Anchoring: Grounding percepts to concepts in cognitive robots*. Doctoral Dissertation.

56. Kristoffersson, Annica (2013) *Measuring the Quality of Interaction in Mobile Robotic Telepresence Systems using Presence, Spatial Formations and Sociometry*. Doctoral Dissertation.

57. Memedi, Mevludin (2014) *Mobile systems for monitoring Parkinson's disease*. Doctoral Dissertation.

58. König, Rikard (2014) *Enhancing Genetic Programming for Predictive Modeling*. Doctoral Dissertation.

59. Erlandsson, Tina (2014) *A Combat Survivability Model for Evaluating Air Mission Routes in Future Decision Support Systems*. Doctoral Dissertation.

60. Helldin, Tove (2014) *Transparency for Future Semi-Automated Systems. Effects of transparency on operator performance, workload and trust*. Doctoral Dissertation.

61. Krug, Robert (2014) *Optimization-based Robot Grasp Synthesis and Motion Control.* Doctoral Dissertation.

62. Reggente, Matteo (2014) *Statistical Gas Distribution Modelling for Mobile Robot Applications*. Doctoral Dissertation.

63. Längkvist, Martin (2014) *Modeling Time-Series with Deep Networks*. Doctoral Dissertation.

64. Hernández Bennetts, Víctor Manuel (2015) *Mobile Robots with In-Situ and Remote Sensors for Real World Gas Distribution Modelling*. Doctoral Dissertation.

65. Alirezaie, Marjan (2015) *Bridging the Semantic Gap between Sensor Data and Ontological Knowledge*. Doctoral Dissertation.

66. Pashami, Sepideh (2015) *Change Detection in Metal Oxide Gas Sensor Signals for Open Sampling Systems*. Doctoral Dissertation.

67. Lagriffoul, Fabien (2016) *Combining Task and Motion Planning.* Doctoral Dissertation.

68. Mosberger, Rafael (2016) *Vision-based Human Detection from Mobile Machinery in Industrial Environments*. Doctoral Dissertation.

69. Mansouri, Masoumeh (2016) *A Constraint-Based Approach for Hybrid Reasoning in Robotics*. Doctoral Dissertation.

70. Albitar, Houssam (2016) *Enabling a Robot for Underwater Surface Cleaning*. Doctoral Dissertation.

71. Mojtahedzadeh, Rasoul (2016) *Safe Robotic Manipulation to Extract Objects from Piles: From 3D Perception to Object Selection.* Doctoral Dissertation.

72. Köckemann, Uwe (2016) *Constraint-based Methods for Human-aware Planning*. Doctoral Dissertation.

73. Jansson, Anton (2016) *Only a Shadow. Industrial Computed Tomography Investigation, and Method Development, Concerning Complex Material Systems.* Licentiate Thesis.

74. Sebastian Hällgren (2017) *Some aspects on designing for metal Powder Bed Fusion.* Licentiate Thesis.

75. Junges, Robert (2017) *A Learning-driven Approach for Behavior Modeling in Agent-based Simulation.* Doctoral Dissertation.

76. Ricão Canelhas, Daniel (2017) *Truncated Signed Distance Fields Applied To Robotics*. Doctoral Dissertation.

77. Asadi, Sahar (2017) *Towards Dense Air Quality Monitoring: Time-Dependent Statistical Gas Distribution Modelling and Sensor Planning*. Doctoral Dissertation.

78. Banaee, Hadi (2018) *From Numerical Sensor Data to Semantic Representations: A Data-driven Approach for Generating Linguistic Descriptions*. Doctoral Dissertation.

79. Khaliq, Ali Abdul (2018) *From Ants to Service Robots: an Exploration in Stigmergy-Based Navigation Algorithms*. Doctoral Dissertation.

80. Kucner, Tomasz Piotr (2018) *Probabilistic Mapping of Spatial Motion Patterns for Mobile Robots*. Doctoral Dissertation.

81. Dandan, Kinan (2019) *Enabling Surface Cleaning Robot for Large Food Silo*. Doctoral Dissertation.

82. El Amine, Karim (2019) *Approaches to increased efficiency in cold drawing of steel wires*. Licentiate Thesis.

83. Persson, Andreas (2019) *Studies in Semantic Modeling of Real-World Objects using Perceptual Anchoring*. Doctoral Dissertation.

84. Jansson, Anton (2019) *More Than a Shadow. Computed Tomography Method Development and Applications Concerning Complex Material Systems*. Doctoral Dissertation.

85. Zekavat, Amir Reza (2019) *Application of X-ray Computed Tomography for Assessment of Additively Manufactured Products*. Doctoral Dissertation.

86. Mielle, Malcolm (2019) *Helping robots help us—Using prior information for localization, navigation, and human-robot interaction*. Doctoral Dissertation.

87. Grosinger, Jasmin (2019) *On Making Robots Proactive*. Doctoral Dissertation.

88. Arain, Muhammad Asif (2020) *Efficient Remote Gas Inspection with an Autonomous Mobile Robot*. Doctoral Dissertation.

89. Wiedemann, Thomas (2020) *Domain Knowledge Assisted Robotic Exploration and Source Localization*. Doctoral Dissertation.